

INTELIGENCIA ARTIFICIAL

Código máquina a tu alcance

El juego de Gusánez



Portada: Luis Gala

Director: Simeón Cruz

12 AÑO I.









Poco a poco se va consiguiendo que los ordenadores hablen, pero todavía no nos entienden, a pesar de cuanto se ha escrito y mitificado sobre la Inteligencia Artificial. Con este tema abrimos este número de agosto, antes de pasar a «zambullirnos» en los temas fuertes de Spectrum.
Y a propósito de «zambullidas», no deje a Gusánez (Gusi para los amigos) sin la suya. Un programa muy especial para los momentos de asueto en este verano.
¡Felices vacaciones!

ACTUALIDAD

10 REM HARDWARE. El mercado de periféricos y programas para Spectrum ha evolucionado desde la aparición de aquellas primeras expansiones externas y joysticks. Asistimos en estos momentos a una situación problemática en el mercado de software: Muchas pequeñas tiendas han abandonado el mercado debido a la continua renovación de productos, que hace muy difícil mantenerse al día sin acumular grandes stocks de programas de dudosa venta.

El mercado de periféricos parece atravesar un buen momento: los productos mejoran de calidad con cada nuevo lanzamiento, y los precios se mantienen estables o, en muchos casos, disminuyen.

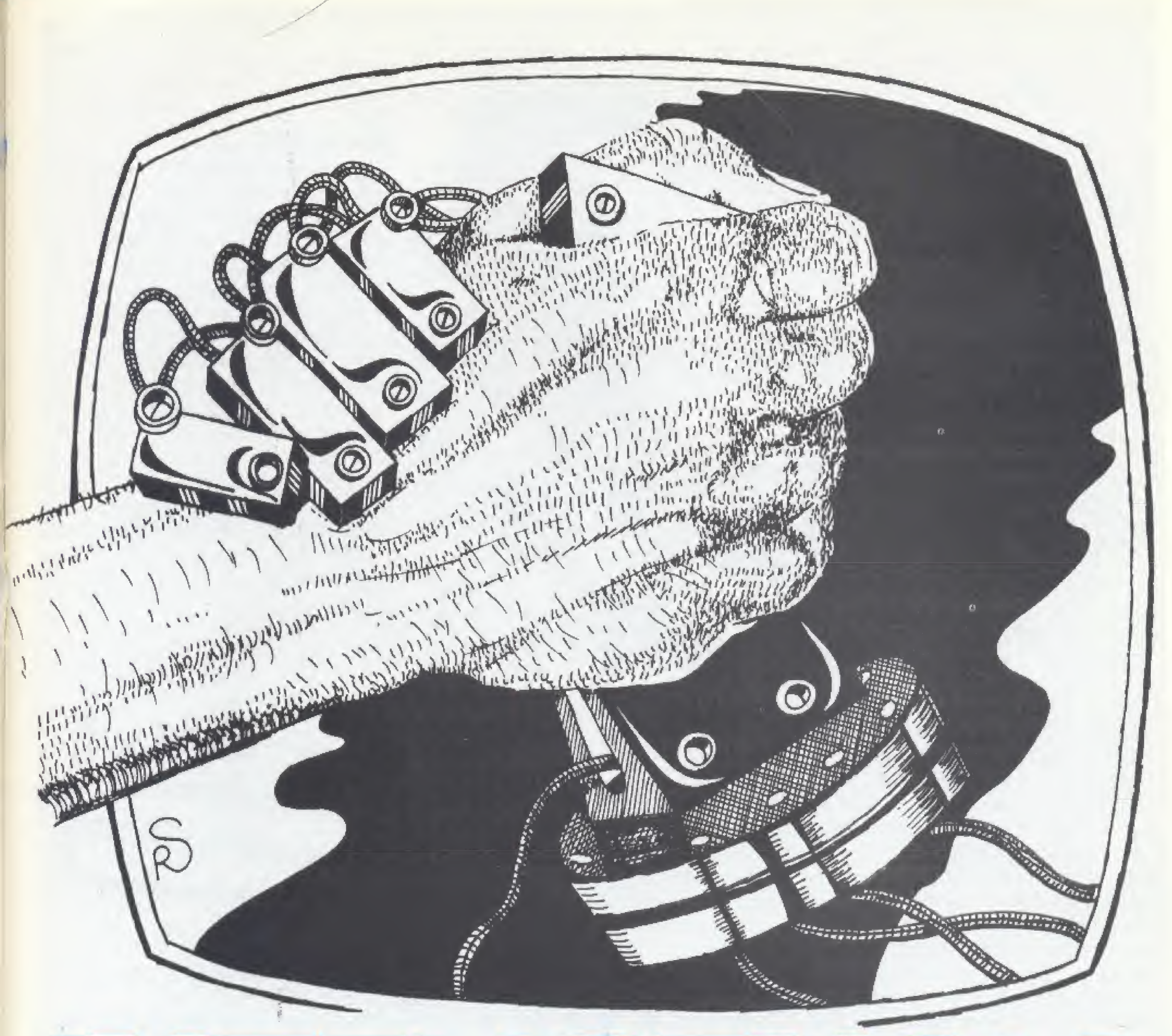
Por ejemplo, el reciente lanzamiento en España del sistema de disco DISCOVERY I, de Technology Research, añade un nuevo producto a la parte alta de la gama de periféricos para Spectrum.

Por otro lado, hemos sabido que Indescomp comercializará a partir de Septiembre una versión muy mejorada del teclado MULTIFUNCION I, ya veterano entre los teclados, y un nuevo Interface para poder jugar con dos joysticks.

También Pin-soft nos comenta que el lápiz óptico que presentamos en el número pasado está funcionando muy bien en el mercado.

20 REM EL QL ESPAÑOL LLEGA A ESPAÑA. Aunque parezca una paradoja, las primeras unidades de esta máquina cruzaron el Canal de la Mancha y están siendo distribuidas. Pudimos conseguir una de ellas, y el mes que viene comentaremos las mejoras que introduce.

30 REM Y la noticia más importante es que Agosto es un mes de vacaciones para la mayoría. ¿No habeis notado que el Spectrum se calienta mucho más de lo normal? Dejad ese programa que no funciona y nos veremos en Septiembre.



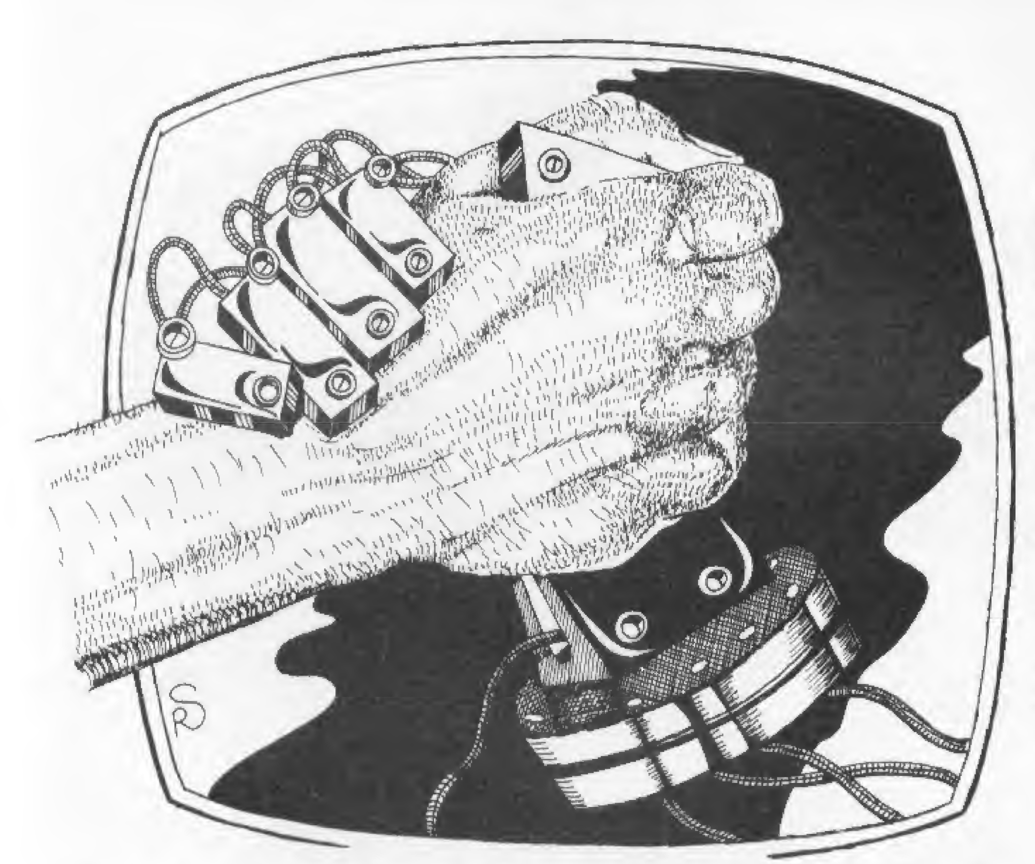
INTELIGENCIA ARTIFICIAL

Clave para entender el pensamiento

La inteligencia artificial tiene como materia de estudio algunas de las más inquietantes preguntas que pueden plantearse: ¿Qué hace nuestra mente cuando pensamos, sentimos, vemos o comprendemos? ¿Es posible averiguar realmente cómo funciona el cerebro? Estas preguntas se repiten desde hace miles de años sin obtener respuesta.

Sin embargo, quienes investigan estos temas disponen ahora de una nueva herramienta: el ordenador. Cualquiera que los utilice puede atestiguar que a menudo dan más problemas de los que solucionan, pero en el estudio de la mente y el pensamiento constituyen un banco de pruebas idóneo.

Las teorías sobre el pensamiento frecuentemente toman la forma de descripción de un proceso. Por ejemplo, una posible explicación al proceso de elaboración de respuestas a preguntas sería: en primer lugar, se transforma la pregunta en una representación interna. Esta se emplea como un índice que busca la zona de memoria ade-



cuada. Su contenido se traduce al formato necesario para una respuesta y por último, se generan las palabras para comunicarla.

Las teorías de procesos parecen un buena manera de describir lo que sucede en el interior del cerebro, pero realmente no aclaran gran cosa. ¿Cómo se organiza la memoria?, ¿puede llegar a saturarse?, ¿cómo se localiza un determinado suceso en la memoria? Un investigador que intentase crear un programa para responder preguntas tendría que enfrentarse a estos problemas y a otros muchos. La programación nos fuerza a ser explícitos y esta es la mejor manera de confrontar las teorías con la realidad.

No hace mucho, los investigadores de la inteligencia artificial enfocaban su actividad sobre lo que consideraban manifestaciones de comportamiento altamente inteligente: jugar al ajedrez, demostrar teoremas matemáticos, etc. Muchos encontraron potentes técnicas de programación para acometer esas tareas «inteligentes». Pero también descubrieron que los procedimientos usados por las personas eran muy diferentes. Actualmente concentran sus energías en la investigación de trabajos que la mayoría consideramos triviales: utilización del lenguaje, aprendizaje partiendo de las experiencias pasadas y lo que llamamos «sentido común».

El lenguaje

Se comenzó con intentos de escribir programas que manejaran de un modo práctico el inglés (y otros idiomas naturales). Se pretendía que pudieran entender el lenguaje escrito lo suficientemente bien como para hacer resúmenes, traducir a otros idiomas o contestar preguntas sobre textos analizados. Pero los investigadores se encontraron de inmediato con una dificultad adicional: las personas entienden normalmente más de una frase que su estricto contenido léxico. Por ejemplo, casi todos estarían de acuerdo en que la frase «Pepe se compró un coche nuevo» contiene una referencia a dinero, aunque esta palabra no aparezca. Cualquier programa que comprenda la frase debe responder «Sí» a la pregunta «¿Se gastó dinero Pepe?».

¿Cómo puede saber esto un programa? Una primera aproximación sería asociar «dinero» a «comprar», declarando que comprar significa «dar dinero a cambio de». Este método emplea los significados de varias palabras de la frase para construir una representación del significado de toda la expresión. Más o menos es un buen sistema, pero tiene algunos inconvenientes. Las palabras son frecuentemente ambiguas y las deducciones que se pueden hacer sobre su sentido fácilmente pueden ser equívocas. La ambigüedad está tan generalizada que cualquier programa que trabaje con lenguajes naturales debe tener poderosos métodos para evitarlas.

Otro problema habitual es que los contenidos «entre líneas» no se deben a ninguna palabra en particular. Por ejemplo, en la siguiente historia: «Pepe fue a un restaurante y pidió un sandwich. El camarero se lo sirvió rápidamente. En agradecimiento le dio una buena propina». Es obvio que Pepe se comió el sandwich y lo pagó, pero estas acciones no se desprenden de ninguna palabra de la historia. ¿Podría deducirlas un programa de ordenador? Para ello debería saber lo que ocurre en los restaurantes. Una persona también podría afirmar que probablemente Pepe se sentó a la mesa, miró la carta, un cocinero hizo el sandwich, etc. El programa necesita poder rellenar los «huecos» no mencionados en el texto.

Clave para entender el pensamiento

Hay una solución para ambos problemas: que el programa genere constantemente previsiones o expectativas sobre lo que ocurrirá inmediatamente después. Las personas raramente advierten la ambigüedad al hablar, porque el significado exacto queda aclarado por el contexto. Un programa que resolviese la ambigüedad tendría que construirse un contexto. En otras palabras, debería presuponer lo que puede ocurrir basándose en lo

sucedido previamente y en lo que ya conozca sobre este tipo de situaciones.

Estas expectativas o previsiones surgen generalmente en bloques y ayudan también a tapar los «huecos» de los que hablábamos anteriormente. En una conversación sobre un restaurante se hace referencia a una variedad de objetos, situaciones y personajes: el cliente observará la carta, pedirá algo al camarero, esperará un rato hasta que le sirvan, comerá, pagará la nota, dejará una propina y se marchará. Es obvio en este ejemplo que los bloques de expectativas pueden ordenarse temporalmente. Cuando una de ellas se cumple, el programa puede deducir que la anterior probablemente también lo hizo. Del mismo modo, las personas saben la secuencia en que se desarrollan las acciones habituales y utilizan ese conocimiento para generar expectativas. No hay forma de entender el lenguaje sin conocer algo del tema que se discute.

Para realizar programas con los que verificar estas ideas habrá que imaginar qué tipo de expectativas existen y cómo van a usarse. Luego habrá que decidir la forma en que las manejará el programa. En cierto modo, todos los analizadores sintácticos de los lenguajes naturales usan las expectativas. La diferencia entre un analizador tradicional, estrictamente sintáctico, y un analizador más conceptual radica en la procedencia de sus expectativas. El primero utiliza sólo conocimientos gramaticales, mientras el segundo incorpora además información de otras fuentes. El analizador tradicional ve la sintaxis como proceso preliminar para otros análisis. El conceptual la ve únicamente como una de las muchas fuentes de información que utiliza simultáneamente para comprender textos.

Muchas de las expectativas «de bajo nivel» proceden del vocabulario utilizado para expresar un significado. Existen programas que usan la dependencia conceptual para representar significados. En su forma básica soportan una acción, un sujeto, un objeto y un componente direccional (desde o hacia). Por ejemplo, la acción «comer» requiere que el objeto sea comestible y que el sujeto esté vivo. Cuando en un texto aparece una palabra que implica comer, se crean expectativas sobre un objeto comestible (generalmente se encuentra tras la palabra «comer») y un sujeto con vida (normalmente precediéndola). Se advierte que las expectativas generadas son tanto semánticas (presuponen el significado de la palabra) como sintácticas (presuponen dónde debe encontrarse la palabra). Por supuesto, incluso las acciones de dependencia conceptual son susceptibles de ambigüedad. Un modo de evitarla sería crear expectativas referentes a los dos significados posibles y escoger la acción cuyas previsiones se cumplan en las restantes palabras del texto. En ese instante, el programa desactivaría las expectativas no satisfechas.

Un programa inteligente debe comprender los sobreentendidos de nuestro lenguaje

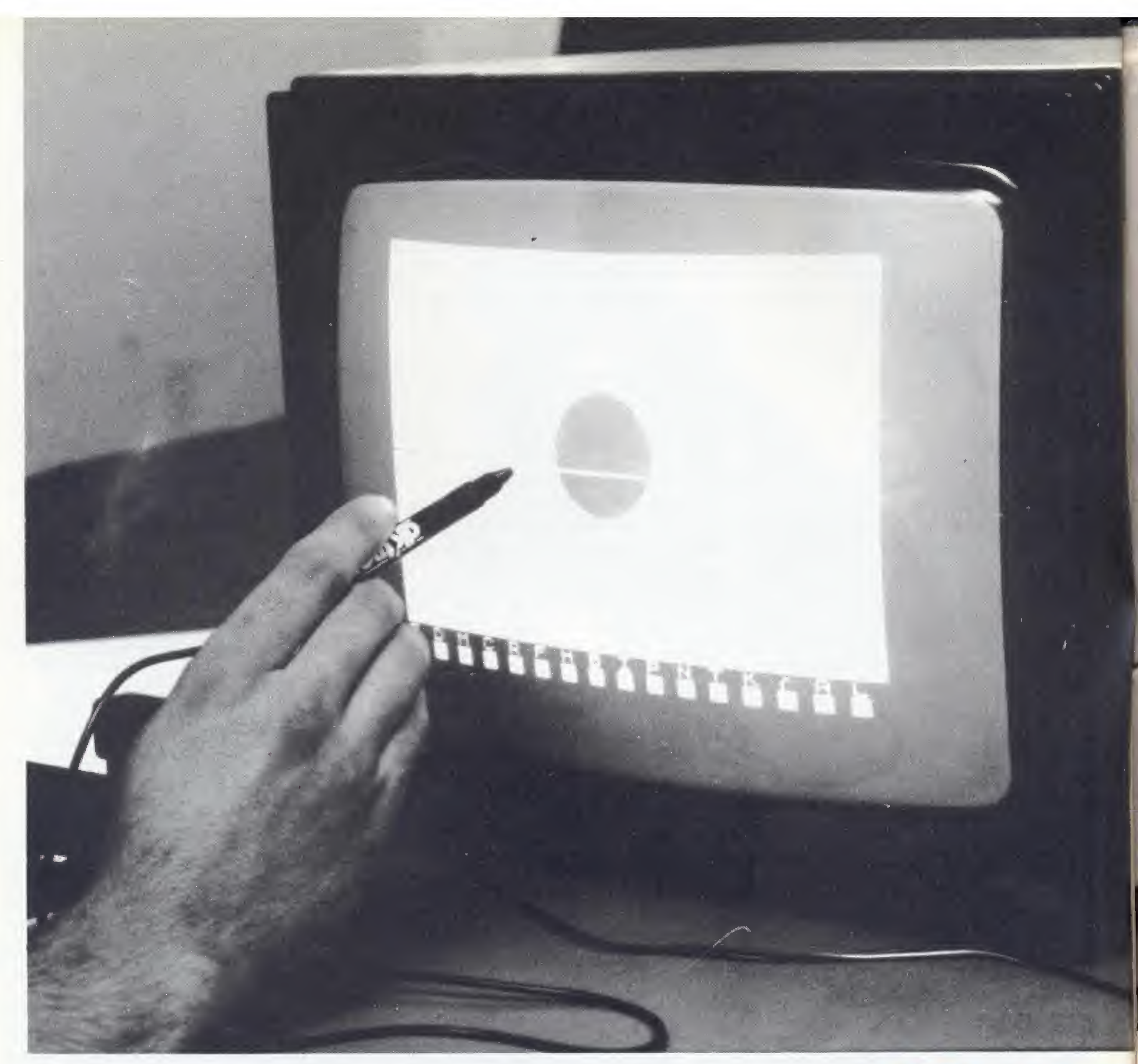
Hay otros modos de eliminar la ambigüedad. Los esquemas de expectativas más complejos pueden proporcionar mejor información. La historia del restaurante contenía una expresión ambigua: «El camarero le sirvió rápidamente. En agradecimiento le dio una buena propina». El pronombre «le» no se refiere al antecedente más reciente, el camarero, sino a otro más remoto: Pepe. La única posibilidad de aclarar el significado de este pronombre en la frase es por la previsión de que los clientes dejan propina a los dependientes. Puede intentarse asociar esta información con el significado apropiado de «propina». Pepe puede actuar como cliente en este episodio, pero en cualquier otra situación podría hacer el papel de dependiente. Las presuposiciones cliente/dependiente/propina y la identificación de Pepe como cliente surgiría de un bloque de expectativas sobre restaurantes.

Cuando los bloques de expectativas comenzaron a utilizarse en los programas, se descubrieron algunos interesantes detalles sobre su funcionamiento. Una historia puede referirse a muchos bloques y éstos pueden relacionarse entre sí de múltiples maneras. Pueden ocurrir a la vez (por ejemplo, comer y viajar en avión) o ser incompatibles (conducir un coche e ir en avión). También pueden decidir el sentido de una determinada palabra asociada a ellos, lo que no resuelve la ambigüedad pero constituye una valiosa ayuda.

Los bloques de expectativas también dan problemas. ¿Cómo puede un programa imaginarse qué bloque utilizar en un momento determinado? Lógicamente, añadiendo condiciones de activación que especifiquen en qué situaciones debe seleccionarse cada uno de ellos. Los primeros programas simplemente buscaban en cada bloque para comprobar si se habían encontrado las condiciones. Pero al aumentar el número de bloques por programa hubo que perfeccionar su organización. Se necesita un método para encontrar el adecuado sin tener que mirarlos todos.

Otro problema es seleccionar lo que debe figurar en cada bloque. Comer en casa tiene mucho en común con comer en un restaurante. ¿Las expectativas de ambas acciones deben ir en el mismo bloque o en dos diferentes? ¿Que se haría con la información relativa a los distintos tipos de restaurantes? Hace falta una teoría sobre cómo organizar y encontrar los grupos de expectativas.

En resumen, lo que empezó como estudio del lenguaje se ha convertido en una investigación sobre el conocimiento y la manera en que éste se organiza. En otras palabras, un estudio de la memoria.



AT

EL lápiz óptico DK'TRONICS, distribuido en España por **ABC Analog,** consta, además del lápiz en sí y su correspondiente *interface,* de un programa de dibujo y un manual de instrucciones, este último bastante completo pero escrito (¡cómo no!) en su lengua nativa, es decir en inglés. Comienza este manual explicando muy someramente lo que es un lápiz óptico: un dispositivo que detecta la luz emitida por el televisor y envía una señal al ordenador, el cual podrá, si dispone del *software* adecuado, localizar el punto de la pantalla sobre el que está situado el lápiz.

Para poder utilizar este periférico deberemos, antes de nada, conectar la clavija del lápiz a la entrada correspondiente en el *interface* y éste al bus trasero de Spectrum. Una vez hecho esto podremos enchufar el ordenador y cargar el programa que se nos suministra tecleando LOAD "".

Ajustando el lápiz a nuestra TV

Si todo ha funcionado bien aparecerá en pantalla un menú con tres alternativas posibles: si pulsamos la tecla «c» saltaremos a la subrutina que ajusta el código máquina a nuestra TV. Para ello deberemos situar el lápiz sobre el cuadro blanco de la izquierda. Aparecerá una flecha a su derecha que deberemos subir (tecla «1») o bajar (tecla «q») para alinearla con la cruz del cuadro. Cuando lo consigamos aparecerá el mensaje *«Correct setting»,* entonces estaremos en condiciones de salvar el código máquina ajustado, pulsando una tecla cualquiera y después la «s». Esto lo podemos hacer en otra cin-



será «en frío», es decir inicializando variables y borrando las pantallas archivadas (caso de que hubiera alguna).

Cuando lo hagamos aparecerán en la parte baja de la pantalla las iniciales de cada uno de los comandos disponibles. Para seleccionarlos deberemos situar el lápiz sobre el cuadro blanco que hay junto a cada uno de ellos y pulsar una tecla cualquiera. En la zona superior veremos los dos cursores que se utilizan para marcar los puntos de la pantalla sobre la que trabajarán los citados comandos. El cursor principal está representado con un «x» y el secundario con las líneas vertical y horizontal que cruzan la pantalla. Para cambiar la posición de este último situaremos el lápiz en el nuevo emplazamiento y pulsaremos una tecla cualquiera; para el principal se utiliza el comando *Move* como explicamos más adelante.

to es aceptable, cómodo y fácil de utilizar. Las posibles opciones son las siguientes:

Erase

Este comando está pensado para borrar la última figura que hayamos dibujado; siempre que ésta haya sido una circunferencia, una recta, un rectángulo o un arco de circunferencia (en realidad lo que hace es repetir el último comando en OVER 1). Desgraciadamente no arreglará nada si nos hemos equivocado al hacer FILL, y si la figura quedaba sobre algo dibujado ya anteriormente esa zona será también borrada. Hay que poner especial atención en no mover los cursores antes de ejecutar esta opción, pues entonces dibujaría la figura en distinto sitio del que estaba la anterior, con lo que el problema será doble; esto, con un poco de habilidad, puede ser utilizado para dibujar en OVER 1, función de la que carece este programa.

SACALE PUNTA U SPECTRUM

Continuamos el estudio de los lápices ópticos disponibles en el mercado español. Este mes nos llega, del otro lado del Canal de la Mancha, otro de estos periféricos, que ofrece el aliciente de poder ser utilizado en nuestros propios programas.

ta tras haber salvado el programa BASIC, para lo cual podremos hacer BREAK, SAVE «lápiz» y RUN para volver al menú.

Programa de dibujo

Aparte de la opción de calibrado, tenemos en el primer menú otras dos opciones que nos permiten acceder al menú principal de programa de dibujo: si pulsamos la tecla «k» lo haremos respetando las pantallas que tengamos almacenadas (ver comando *Keep*) y si pulsamos cualquier otra tecla la entrada

Comandos y opciones

A pesar de que este programa dispone de 16 comandos de dibujo harían falta algunos más que lo completarán para poder hablar de un paquete serio con el que pudiéramos realizar gráficos algo complejos. De todas formas el conjun-

Draw

Dibuja una línea recta entre los dos cursores. Para ejecutar este comando deberemos, una vez que hayamos definido la posición que queremos, poner el lápiz sobre el cuadrado correspondiente a la «D» y pulsar una tecla.



Move

Este comando desplaza el cursor principal al lugar que ocupa el secundario, quedando, por lo tanto, ambos en el mismo punto. Para mover de sitio el cursor secundario basta apuntar el lápiz al punto que deseemos y pulsar una tecla.

Circle

Traza una circunferencia con centro en el cursor principal y cuyo radio será igual a la distancia entre éste y el cursor secundario. En caso de que la circunferencia no coja en la pantalla dará el mensaje de error *«Radius of circle is too big, re-position pen and try again»*, algo así como «El radio de la circunferencia es demasiado grande, vuelve a situar el lápiz y prueba de nuevo».

Rectangle

Dibuja un rectángulo cuyos vértices serán los dos cursores.

Fill

Opción de rellenado, típica en todo buen programa de dibujo. En este caso la rutina que realiza el trabajo no está del todo conseguida. Aparte de ser lenta, no siempre admitirá que tomemos como límites de la superficie a rellenar los bordes de la pantalla; quedándose «colgado», en caso de que lo hagamos, durante bastante tiempo (alrededor de media hora) y dando al final un *Out of memory*.

El Interface resulta más voluminoso que en otros lápices, pero a cambio deja libre la toma EAR.

Para ejecutar este comando deberemos señalar, con el cursor secundario, un punto de la superficie a rellenar y señalando al cuadrado de la letra «F» pulsar una tecla. Una vez hecho esto no podremos echarnos atrás, por lo que conviene asegurarse antes de hacerlo de que la figura que pretendemos rellenar no tiene ningún hueco por donde pueda escapar el «flujo» de tinta.

Hand-draw

Este comando nos da la posibilidad de dibujar en la pantalla como si estuviéramos haciéndolo con un lápiz de los de madera y grafito (o casi). Para ello deberemos marcar con el lápiz el cuadro correspondiente (el «H») tras lo que aparecerá el mensaje *«Press any key to start sketching»*, que en castellano es «Pulsa una tecla para comenzar a dibujar». Para dejarlo bastará con pulsar una tecla de nuevo.

Border, Ink, Paper

Estos tres comandos trabajan de forma similar. Definen el color del BORDER o los atributos en curso (salvo brillo y flash, inexplicablemente ausentes en un programa de dibujo). Cuando marquemos con el lápiz las casillas correspondientes aparecerá en la parte inferior de la pantalla una tira con los ocho colores posibles en el Spectrum. Sólo tendremos que señalar con el lápiz el que deseamos y pulsar una tecla.

New screen

La pantalla es borrada y redefinida con los atributos en curso, los cursores son colocados en el centro.

Tape

Podemos utilizar este comando para salvar o cargar pantallas o desde *cassete* o *microdrive*. Cuando lo ejecutemos aparecerán en la parte inferior de la pantalla tres opciones: SAVE, LOAD y ABORT (para volver al menú). Una vez elijamos SAVE o LOAD aparecerán otras dos opciones: TAPE (cassette) y MICRODRIVE.

Con respecto a otros lápices el Dk'Tronics ofrece la ventaja de poder ser usado en nuestros programas.

Keep

Este comando, que sólo puede ser utilizado en la versión de 48 K, salva el contenido de la pantalla en memoria. Podremos archivar de esta manera hasta cinco pantallas distintas que podrán ser recuperadas usando el comando que sigue.

Recall

Cuando lo ejecutemos se nos mostrarán los números de las pantallas que haya en memoria (caso de haber alguna), junto a la opción CYCLE; si elegimos esta última lo que hará es mostrarnos, a la velocidad que le indiquemos, todas las

YA ESTÁ A LA VENTA



PIDALA EN SU QUIOSCO HABITUAL O SOLICITELA A:

INFODIS, S.A.
Bravo Murillo, 377 • 28020 Madrid

VIRGIN-1-SPECTRUM

CUPON DE PEDIDO

Ruego me envíen la cinta cassette **VIRGIN SOFTWARE** al precio de **525** pts.
El importe lo abonaré: Con mi tarjeta de crédito □ American Express □
Visa □ Interbank □ Adjunto cheque □
Contra reembolso □
Número de mi tarjeta
Fecha de caducidad
NOMBRE
DIRECCION
CIUDAD
C.P. PROVINCIA
Sin gastos de envío



pantallas que en ese momento haya archivadas.

Arc

Traza un arco de circunferencia; para ello deberemos definir tres puntos: los dos extremos serán indicados con los cursores, y otro punto del arco será tomado de la última posición del arco secundario. En caso de que el arco resultante se salga de pantalla el programa se interrumpirá con mensaje: *«B: Out of screen»*. Para continuar deberemos hacer RUN, y pulsar la tecla «K» para que las pantallas que hayamos dibujado sean respetadas.

Letters

Este comando nos permite insertar textos a partir del punto marcado por el cursor secundario. Cuando lo ejecutemos nos pedirá el texto que queremos sea insertado; deberemos teclearlo con normalidad y pulsar ENTER.

Utilizando el lápiz en nuestros programas

La mejor de las opciones que nos ofrece este periférico es la posibilidad de utilizarlo en nuestros propios programas. Para ello aprovecháremos la rutina encargada de localizar las cordenadas del punto donde el lápiz está situado. Dicha rutina deberemos salvarla en cinta con SAVE «nombre» CODE 62564,1706 para la versión de 48 K y SAVE «nombre» CODE 29796,1706 para la de 16 K. Para cargarla en nuestros programas haremos CLEAR 62563 (29795 para 16 K): LOAD «nombre» CODE. Una vez que lo tengamos en memoria podremos localizar el lugar

El programa de dibujo es sencillo, siendo el único lápiz disponible para 16 K.

en donde está situado el lápiz de la siguiente forma: LET coords = USR 62564 (29796 para 16 K): LET y = INT (coords/256): LET x = coords – 256*y.

Cuando ejecutemos esto quedarán las coordenadas en las variables x (horizontal) e y (vertical); a partir de ahí son nuestro ingenio e imaginación quienes tienen que decidir qué aplicación queremos darle a esta interesante posibilidad. Si nos conformamos con conseguir el número de línea sólo tendremos que hacer lo siguiente: LET línea = USR 63109.

El programa adjunto es un ejemplo de cómo podemos confeccionar un menú utilizando este lápiz óptico. Antes de ejecutarlo deberemos haber salvado en cinta la rutina de localización de coordenadas de la forma anteriormente expuesta, y poner el *cassette* en condiciones para cargarla cuando el programa nos lo pide. El programa «testea» la variable de sistema P-RAMT para averiguar si nuestro Spectrum es de 16 o de 48 K, por lo que debe funcionar a la perfección en ambos casos.

No es nada difícil, como hemos visto, utilizar este periférico en nuestros programas o incuso adaptar programas comerciales para que admitan esta posibilidad. Esperamos nos comuniquéis vuestras experiencias en este campo en el que todo es decidirse. Esperamos igualmente que hayan quedado claras las características y modo de empleo de este lápiz óptico y que esto os ayude a saber cuál es el que más se ajusta a vuestras necesidades. En el caso de que las aproximadamente 7.000 pesetas por las que se comercializa os parezca demasiado, y si os consideráis unos perfectos «manitas», aún podéis acudir al montaje del número seis de nuestra revista y construiros vosotros mismos un útil de dibujo que facilite el manejo del Spectrum.

Luis Gala

```
   5 REM   Menu de demostracion
para lapiz Dk'tronics 16 y 48K

  10 PRINT "Carga rutina localiz
acion"
  20 GO SUB 200
  30 CLEAR K-2972
  40 LOAD ""CODE
  45 GO SUB 200
  50 CLS
  60 PRINT "MENU-DEMO:Pon lapiz
sobre opcion y pulsa una tecla"
  70 FOR n=0 TO 9
  80 PRINT : PRINT "Opcion ";n;"
 "; FLASH 1;" "
  90 NEXT n
  95 PAUSE 0
 100 LET linea=USR (K-2426)
 110 FOR n=3 TO 22 STEP 2
 120 IF linea=n THEN  CLS : PRIN
T "Has elegido la opcion ";(n-3)
/2: BEEP .5,20: GO TO 50
 130 NEXT n
 140 GO TO 95
 145
 190 REM  Subrutina calculo
         ultimo octeto RAM fisica

 200 LET K=PEEK 23732+256*PEEK 2
3733: RETURN : REM 16 o 48K
```

SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 • 250 pts.
Cómo usar el microdrive/Programación Basic/Ampliación Basicare/Rutina despertador/Variables del sistema/Entrada datos mediante máscaras/Protección del software/Sintonice su Spectrum/Programas.

Núm. 2 • 250 pts.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 • 250 pts.
Novedades sonimag '84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 • 250 pts.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad /Calendario/Pascal/Programas.

Núm. 5 • 250 pts.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 • 250 pts.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: tortugas para todos/Interrupciones del Z-80/Programas.





DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

Todospectrum

SIN NECESIDAD DE ENCUADERNACION



(cada tapa es para 6 ejemplares)



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 250 pts.

Por favor envíenme .... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 600 pts. más gastos de envío.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta: □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

SHADOW FIRE

Erbe
Spectrum 48 K
2.000 ptas.

El embajador Krixix ha sido capturado por el malvado Zoff. Con él, han sido capturados los planos de Shadowfire, una nave espacial de nuevo tipo. Si Zoff capturase esos planos, el Imperio caería en sus manos en poco tiempo. En pocas horas, los esbirros de Zoff descubrirán los planos, ocultos en un microfilm en el espina dorsal de Krixix. Nuestra misión es evitarlo.

Para ello, el embajador nos ha nombrado jefes del equipo Enigma. Este equipo, formado por seis miembros, deberá penetrar en Zoff V.



Manto teletransporta a sus compañeros a la nave Zoff V.



Silk y thorik están bajo ataque de dos enemigos. Tras ordenar a Thorik que pase al ataque, le indicamos a Silk que le ayude.

Control: Teclado, joystick.
Jugadores: Uno.
Gráficos: Muy buenos, con iconos que simbolizan las opciones en los menús.
Sonido: Unos pitidos que nos avisan cuando somos atacados. No es su fuerte.
Nivel de dificultad: Difícil; no dispone de ninguna opción para dosificar la dificultad.
Originalidad: Primera aventura con entrada mediante menús con selección gráfica.
Conclusión: Pese a la dificultad de localizar al embajador en límite de tiempo que se nos da, el juego resulta atractivo. La mayor prueba es que apuremos ese tiempo en cada partida.

Nuestro equipo está compuesto por tropas de élite, como Zark, humano o Sylk, insectoide. Ambos muestran gran resistencia y fortaleza física. Sevrina, humana y peligrosa delincuente, es experta en abrir cerraduras. Resulta rápida, aunque poco resistente. Torik, aviano y compañero de Sevrina, es también rápido.

Los dos restantes miembros del equipo son dos androides: Manto, especializado en teletransporte, y Maul, diseñado para el transporte de armas. Ambos resultan lentos. Los personajes pueden ir armados, recogiendo las armas en nuestra nave de ataque.

Para cumplir la misión, debemos recorrer la nave hasta encontrar al embajador, y sacarlo de la Zoff V. Los movimientos y las diversas acciones se realizan por selección en un menú, estando indicadas las diversas opciones por «iconos». Si posicionamos el *joystick* sobre una de las opciones y pulsamos el botón de disparo, queda seleccionada esa opción.

La estructura de menús es completa, comenzando por un menú para cada personaje, que incluye tres submenús: el de ataque, el de movimiento y el de recogida de objetos. La originalidad de una aventura sin barreras idiomáticas es una de las características más interesantes de este juego. Un buen diseño gráfico y una velocidad de respuesta buena complementan este juego, que hará pasar buenos ratos a los amantes de la aventura.

MATCH POINT

Investrónica
Spectrum 48 K
2.100 ptas.

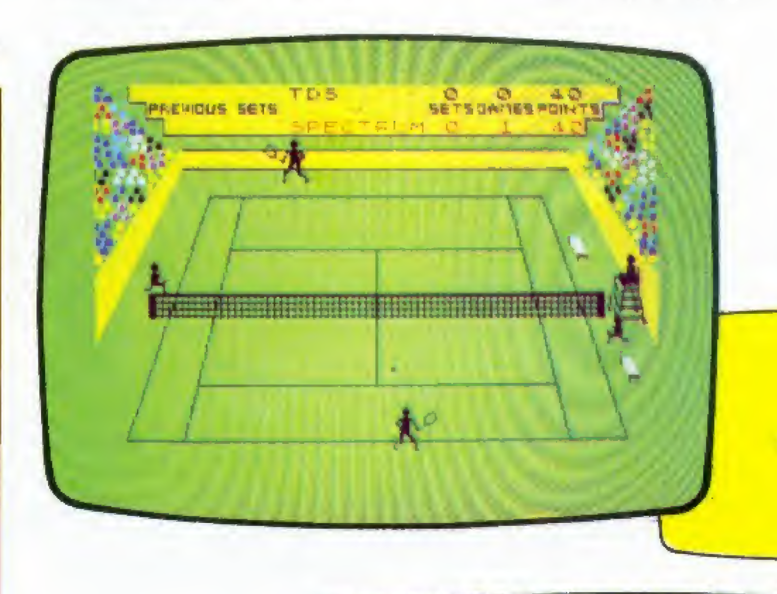
Uno de los primeros juegos de inspiración deportiva que aparecieron para el Spectrum y la realización definitiva del tenis en un ordenador. Aunque se trata de un programa relativamente antiguo, el ambiente relajado de las vacaciones de verano es perfecto para un juego como éste, donde la concentración y rapidez de reflejos resultan imprescindibles.

El juego admite manejo desde teclado o *joystick*, aunque resulta necesario un *Interface* que admita dos palancas si queremos competir en igualdad de condiciones contra un oponente humano. Existen tres niveles de dificultad que se distinguen por la velocidad de la bola y los efectos que se pueden lograr. Se puede dosificar también la duración del partido: un set o partidos al mejor de tres o cinco sets.

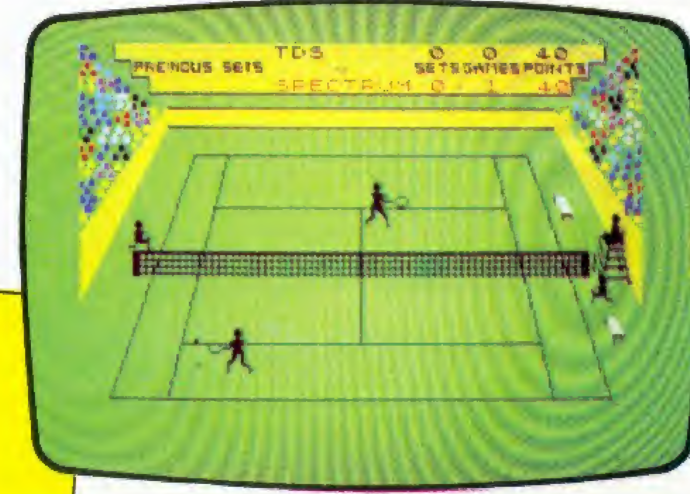
Las reglas del tenis han sido respetadas hasta el *tie-break*, que consiste en que, si en un *set* se llega a empate de seis juegos, la victoria se la apunta el jugador que llegue a siete tantos.

La representación gráfica es correcta, dentro de las posibilidades del Spectrum. Su principal problema es que se tarda en acostumbrarse a la perspectiva, fallando inicialmente muchos golpes por no saber dónde está exactamente la bola. Incluso cuando ya estamos acostumbrados, el juego resulta difícil, necesitando mucha concentración para ganarle a la máquina incluso al nivel más sencillo. Un juego muy adecuado para las vacaciones, siempre que tengamos cerca el mar o la piscina para bañarnos después de las agotadoras partidas.

Control: Teclado, joystick.
Jugadores: Uno o dos.
Gráficos: La presentación 3D del campo de tenis es muy buena. El movimiento es excelente. La animación de los jugadores es también excelente; sólo los recogepelotas dejan algo que desear.
Sonido: No tiene.
Nivel de dificultad: Tres niveles: cuartos de final, semifinal y final. La longitud del partido puede ser de uno, tres o cinco sets.
Originalidad: Aunque existen muchos programas deportivos, muy pocos de ellos incluyen una simulación tan completa. La posibilidad de dos jugadores es algo que también se echa de menos en muchos juegos.
Conclusión: Los programadores del Psion han llevado, como otras veces, las posibilidades de la máquina al límite. Un programa adictivo, donde el joystick resulta indispensable si queremos plantarle cara al ordenador.



Un saque potente es el primer requisito para ganar la partida.



Las subidas a la red y el juego agresivo son elementos imprescindibles de una victoria.

BINGO

El programa que os presentamos a continuación sortea los números de un bingo después de haber preparado los cartones para cada jugador. Para que no todo sea teclear y jugar, el programa es analizado a fondo. Esperamos que los comentarios acerca de su estructura os resulten útiles en vuestros trabajos de programación.



EN el juego del BINGO hay una persona encargada de «cantar» los números entre 1 y 99 de forma totalmente aleatoria y cada jugador tiene un cartón con un conjunto determinado de números (unos 15). Los cartones de los jugadores contienen números diferentes. El ganador es aquel cuyos números han sido cantados en su totalidad.

El programa admite hasta cuatro jugadores y comprueba el cartón ganador.

Procedimiento

Los módulos de programación están dispuestos de la forma siguiente:

1. Prepara el juego.
2. El cuerpo principal del juego.
3. Resultado final.

Las distintas subrutinas que hay dentro del programa son:

1.1. Instrucciones preliminares.

1.2. Prepara los números del bombo.

1.3. Prepara los cartones de los jugadores.

1.4. Imprime los números de cada uno.

2.1. Visualiza los números cantados.

2.2. Muestra (mediante una interrupción) el cartón de los jugadores.

3.1. Lista de los números del cartón.

3.2. Muestra los números cantados.

3.3. Comprueba el cartón ganador.

1.1. Estas son las instrucciones que se necesitan al comenzar el juego. En cada paso del programa es fundamental indicar al jugador la forma de proceder.

1.2. La preparación del bombo requiere una lista de números aleatorios del 1 al 99 (sin repetición) que se almacenarán en un vector que será A(99).

1.3. Los cartones de los jugadores son de 4 tipos con 15 números enteros y totalmente aleatorios, desde el 1 al 99 (números distintos). Cada cartón deberá tener una lista de números y cada lista ha de ser distinta. Estos se almacenarán en los vectores Q(15), R(15), S(15) y T(15).

1.4. Visualiza el cartón del jugador y le da tiempo a que lo anote.

2.1. Muestra, a gran tamaño, los números que se van cantando.

2.2. Interrumpe el juego para comprobar aciertos.

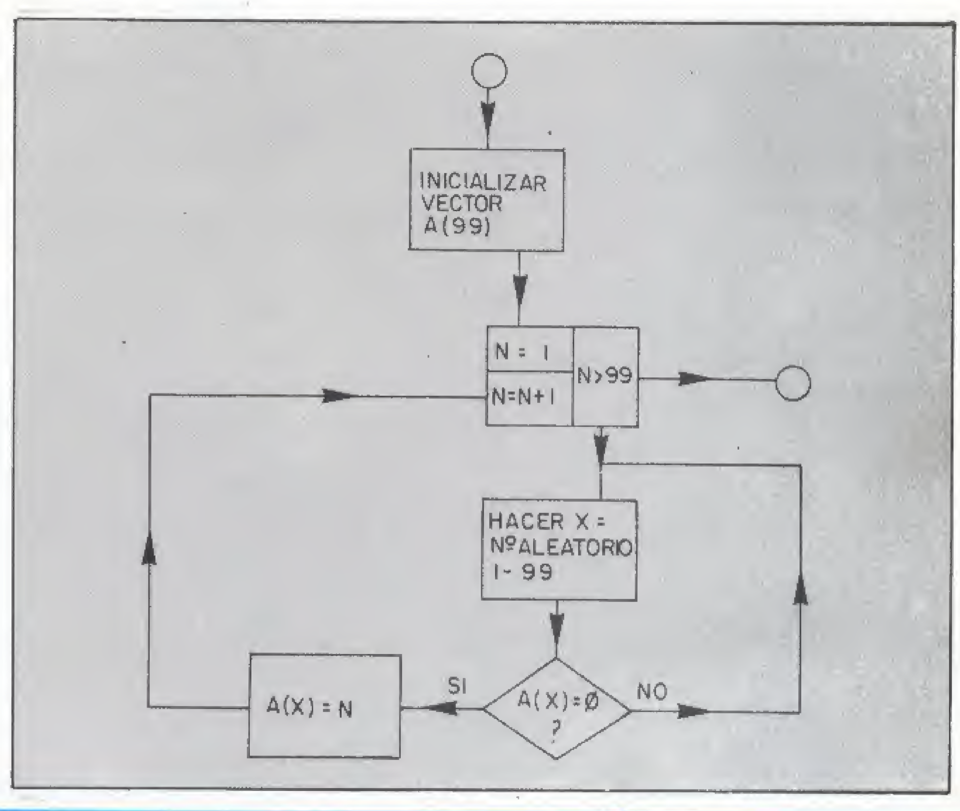
3.1. Repite el punto 1.4, para que los jugadores comprueben sus números.

3.2. Los números cantados se colocan por orden en el vector P(99) y se visualizarán en la pantalla.

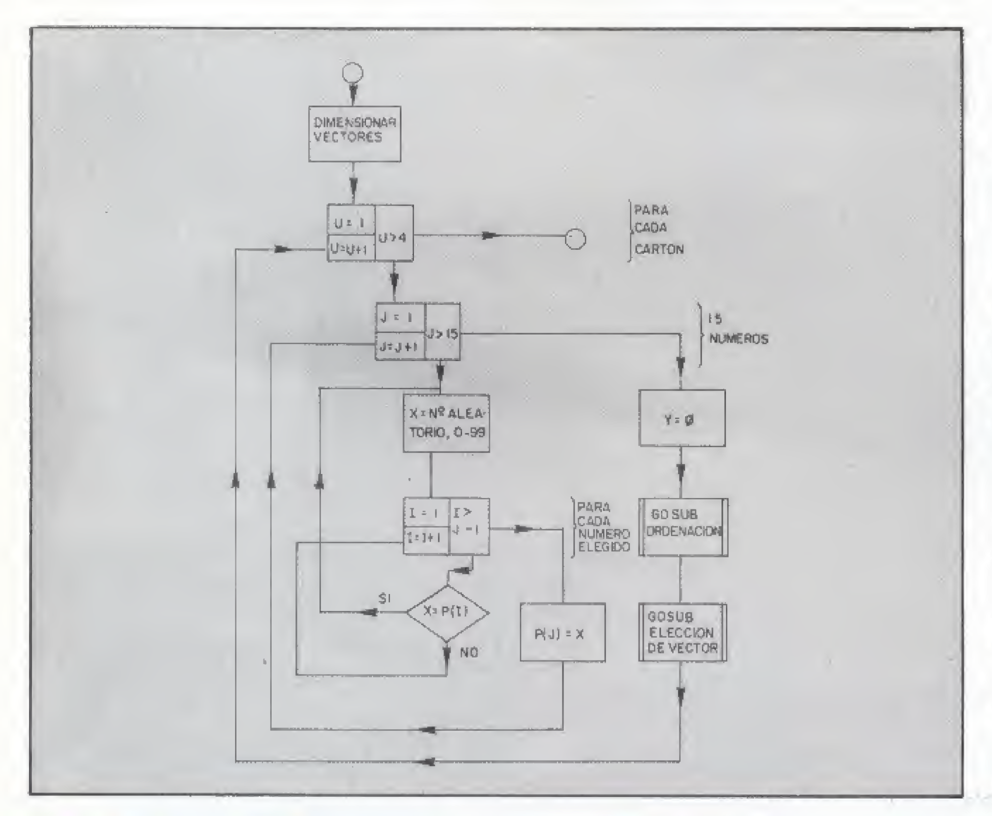
3.3. Se selecciona el cartón ganador y se comprueba. Los números se almacenarán en el vector V(15). Estos se comparan con los que ya han salido, que están almacernados en P(99), y de esa forma se comprueba que esté todo correcto.

Descripción del algoritmo

1.1. Esta sección muestra las instrucciones mínimas para jugar. A lo mejor desea más instruccio-




```
   5 REM "BINGO"
  10 REM ****************
     1.1  * INSTRUCCIONES *
     ===  ****************
  20 PRINT TAB 10;"BINGO";TAB 10
;"*****"
  30 PRINT '"VOY A GENERAR 4 CAR
TONES DE"
  40 PRINT "BINGO CON LAS LETRAS
 'Q', 'R',  'S' Y 'T'."
  50 PRINT '"SE COPIARA EN LA IM
PESORA, SI LAHAY. DE LO CONTRARI
O DEBERAS"
  60 PRINT "ANOTAR EL CARTON Y L
A LETRA      CORRESPONDIENTE DEL
ELEGIDO."
  70 PRINT '"HABRA UN RETARDO AN
TES DE EMPE- ZAR YA QUE ESTARE P
REPARANDO LOSCARTONES."
  80 PRINT '''''"PULSA UNA TECLA
 CUANDO ESTES      PREPARADO."
 110 IF INKEY$="" THEN  GO TO 11
0
 120 CLS
 200 RANDOMIZE
 300 REM ********************
     1.2  *PREPARANDO EL BOMBO*
     ===  ********************
 310 GO SUB 2500
 320 REM ********************
     1.3  *PREPARANDO CARTONES*
     ===  ********************
 420 DIM Q(15)
 430 DIM R(15)
 440 DIM S(15)
 450 DIM T(15)
 460 DIM P(99)
 470 FOR U=1 TO 4
 480 FOR J=1 TO 15
 490 LET X=INT (99*RND)+1
 500 FOR I=1 TO J-1
 510 IF X=P(I) THEN  GO TO 490
 520 NEXT I
 530 LET P(J)=X
 540 NEXT J
 550 LET Y=15
 560 GO SUB 3000
 570 GO SUB 3500
 580 NEXT U
 590 PRINT
 600 PRINT
 610 REM **********************
     1.4  * MOSTRAR LOS NUMEROS *
     ===  **********************
 620 GO SUB 1600
 630 REM ****************
         * COPIAR NUMEROS *
         *SI HAY IMPRESORA*
         ****************
```



nes, para lo cual deberá alterar las líneas 10 a 110 del programa.

1.2. Prepara los números del bombo. Observar el diagrama de flujo.

Esta rutina empieza en la línea 120, limpiando la pantalla y está contenida dentro de la subrutina que prepara los números a cantar (líneas 2500 a 2590). Hay que tener en cuenta que puede ser una secuencia de una subrutina dentro de la parte principal del programa, ya que se ejecuta sólo una vez.

1.3. Prepara los cartones del juego. Veamos el diagrama siguiente.

En el programa esta rutina empieza en la línea 320. Observar el vector P(99), que se utiliza en la subrutina sort y a su vez en la rutina de los números del cartón.

Las líneas 480 a 540 seleccionan los 15 números aleatorios. La su-

```
 640 COPY
 650 PRINT ''"PULSA UNA TECLA PA
RA CONTINUAR."
 660 PAUSE 0
 670 CLS
 680 PRINT "SE VISUALIZARAN LOS
NUMEROS UNO A UNO."
 690 PRINT
 700 PRINT "CUANDO COMPLETES EL
ARTON, PULSAUNA TECLA CUANDO EL
NUMERO ESTE EN LA PANTALLA."
 710 PRINT
 720 PRINT "PULSA UNA TECLA PARA
 CONTINUAR."
 730 IF INKEY$="" THEN  GO TO 73
0
 735 CLS
 740 REM ********************
    2.1  * NUMEROS CANTADOS *
    ===  ********************
 745 LET Z=1
 750 DIM B$(99,2)
 755 CLS
 760 PRINT "PULSA UNA TECLA CUAN
DO TU CARTONESTE COMPLETO."
 770 FOR N=Z TO 99
 780 GO SUB 5000
 790 LET P(N)=A(N)
 795 PAUSE 200
 800 FOR F=1 TO 400
 805 IF INKEY$<>"" THEN  GO TO 8
30
 810 NEXT F
 815 LET Z=Z+1
 820 NEXT N
 830 CLS
 840 FOR N=1 TO 10
 850 PRINT TAB N;"<*BINGO*><*BIN
GO*>"
 860 NEXT N
 870 PRINT ,,"CHECK YOUR CARDS"
 880 REM ********************
         *MOSTRAR CARTON DE *
    2.2  *  LOS  JUGADORES  *
    ===  ********************
 890 PAUSE 100
 900 CLS
 910 LET Y=Z
 920 PRINT "TUS NUMEROS FUERON"
 930 PRINT
 935 REM ****************
         *LISTAR NUMEROS*
    3.1  *  DEL CARTON  *
    ===  ****************
 940 GO SUB 1600
```

brutina 3000 a 3170 se encarga de intercalarlos.

La subrutina 3500 a 3560 coloca los números en el vector adecuado.

1.4. Esta subrutina (líneas 1600 a 1650) visualiza los números en los cartones en forma de tabla.

2.1./2.2. Muestra los números cantados y se interrumpe si se ha pulsado una tecla.

Esta rutina toma los números que se generan, los visualiza a 16 veces su tamaño normal y lo almacena en P. La variable Z cuenta cuántos números han salido. Al final de esta subrutina, los números cantados se almacenarán en la cadena P (que contendrá Z números).

3.1. Repite el punto 1.4.

3.2. Esta rutina (línea 960 a 1070) llama a una pequela subrutina en la línea 3000 que intercalará la lista P(Z) en orden numérico y los imprimirá en la pantalla.

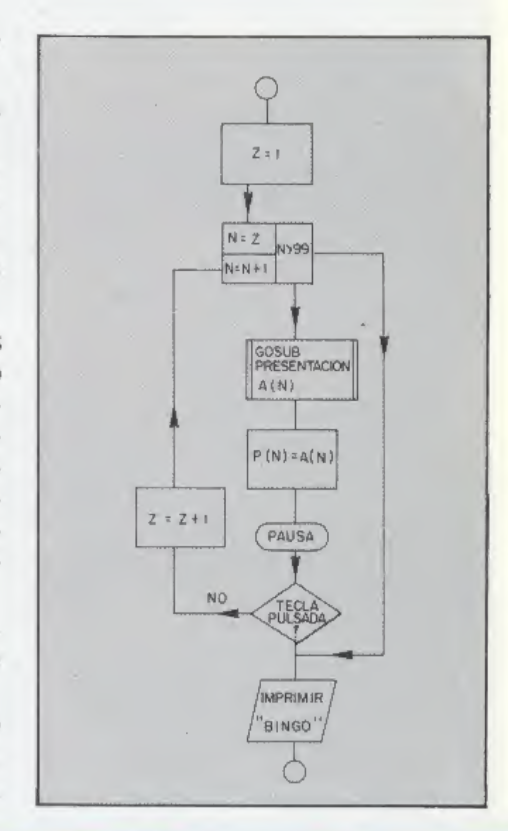
3.3. Esta rutina (líneas 4000 a 4260) se divide en dos partes.

(1) Selecciona el cartón ganador y prepara la lista de los números contenidos en ella para almacenarlos en la cadena V(15) (líneas 4000 a 4120).

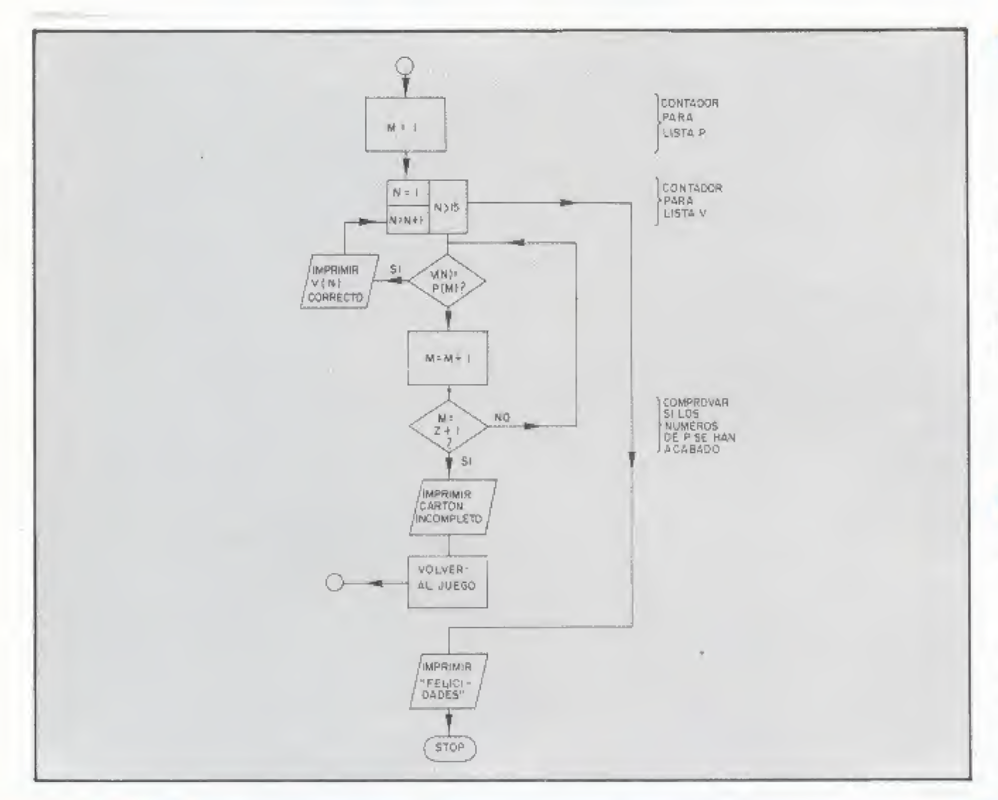
(2) Comprueba los números del cartón ganador. Es necesario buscar la lista ordenada P conteniendo los Z números para compararlos con la lista ordenada V conteniendo 15 números. Si falta cualquier número, el cartón estará incompleto. Si están todos los números se visualizará una felicitación.

El método más rápido para buscar los números es de la siguiente manera:

Se busca por V(1) en la lista P hasta que se encuentra, por ejemplo P(12). A continuación, se bus-




```
 950 REM ****************
         *MOSTRAR NUMEROS*
    3.2  *    CANTADOS   *
    ===  ****************
 960 PRINT "PULSA UNA TECLA PARA
 COMPROBAR  LOS NUMEROS CANTADOS
."
 970 PRINT "TARDARE UN POCO EN M
EZCLARLOS."
 980 PAUSE 0
 990 CLS
1010 GO SUB 3000
1020 CLS
1030 PRINT "NUMEROS CANTADOS"
1040 PRINT
1050 FOR N=1 TO Z
1060 PRINT P(N);" ";
1070 NEXT N
1080 PRINT
1085 REM *****************
         *COMPROBAR CARTON*
    3.3  *    GANADOR     *
    ===  *****************
1090 GO SUB 4000
1100 LET Z=Z+1
1110 PRINT ''"DESEAS CONTINUAR C
ON EL JUEGO?  (S/N)"
1120 INPUT C$
1130 IF C$="N" THEN  STOP
1140 IF C$="S" THEN  GO TO 755
1150 PRINT "POR FAVOR, INTRODUCI
R 'S' O 'N'."
1160 GO TO 1120
1190 REM *******************
         * FIN DEL PROGRAMA *
         *******************
1200 STOP
1580 REM **************
         * SUBRUTINAS *
         **************
1590
1600 REM ***********************
         *SUBRUTINA PARA LISTAR*
         *LOS CARTONS DEL JUEGO*
         ***********************
1610 PRINT "Q";TAB 6;"R";TAB 12;
"S";TAB 18;"T"
1620 FOR J=1 TO 15
1630 PRINT Q(J);TAB 6;R(J);TAB 1
2;S(J);TAB 18;T(J)
1640 NEXT J
1650 RETURN
1660 REM *******************
         * FIN DE SUBRUTINA *
         *******************
1670
```



Ya no habrá que agacharse para sacar las bolas debajo del sofá; con el Spectrum los números no se salen del bombo.

ca V(2) en la lista P a partir de P(13), etc.

La búsqueda finalizará en cuanto un número de la lista V no esté en la lista P pero continuará mientras encuentre los números de V en P. El siguiente diagrama ilustrará este paso.

```
2500 REM *********************
         *RUTINA PARA PREPARAR*
         *LOS NUMEROS A CANTAR*
         *********************
2530 DIM A(99)
2540 FOR N=1 TO 99
2550 LET X=INT (99*RND)+1
2560 IF A(X)<>0 THEN  GO TO 2550
2570 LET A(X)=N
2580 NEXT N
2590 RETURN
2600 REM ********************
         * FIN DE SUBRUTINA *
         ********************
2700
2990 REM ********************
         * SUBRUTINA DE SORT *
         ********************
3000 LET S=Y
3010 LET S=INT (S/2)
3030 IF S>=1 THEN  GO TO 3050
3040 GO TO 3170
3050 FOR K=1 TO S
3060 FOR A=K TO Y-S STEP K
3070 LET B=A
3080 LET T=P(A+S)
3090 IF T>=P(B) THEN  GO TO 3130
3100 LET P(B+S)=P(B)
3110 LET B=B-S
3120 IF B>=1 THEN  GO TO 3090
3130 LET P(B+S)=T
3140 NEXT A
3150 NEXT K
3160 GO TO 3010
3170 RETURN
3180 REM ********************
         * FIN DE SUBRUTINA *
         ********************
3190
3490 REM *****************
         *SUBRUTINA PARA 4*
         *ARRAYS NUMERICOS*
         *****************
3500 FOR J=1 TO 15
3510 IF U=1 THEN  LET Q(J)=P(J)
3520 IF U=2 THEN  LET R(J)=P(J)
3530 IF U=3 THEN  LET S(J)=P(J)
3540 IF U=4 THEN  LET T(J)=P(J)
3550 NEXT J
3560 RETURN
3570 REM ********************
         * FIN DE SUBRUTINA *
         ********************
3580
```

Tabla de datos

Se han utilizado las siguientes variables:

Q(15), R(15), S(15), T(15) son tablas que contienen los números de los jugadores.

A(99) es el vector que contiene los números a cantar.

P(99) es la tabla que contiene los números que se han cantado y también se ha utilizado como almacenamiento temporal para preparar los números de los jugadores y en la rutina de sort.

X es un número aleatorio entre 1 y 99.

Z es el número de números cantados.

Y es el número de elementos que hay actualmente en el vector a intercalar.

Comentamos exhaustivamente el programa, para facilitar su mejora o modificación.

V(15) es el vector que contiene la lista de números ganadores.

B$(99,2) es una cadena que contiene los valores correspondientes a los elementos en P(99).

C es el código del caracter del segundo caracter en el array B$.

D es el código del caracter del primer caracter en el array B$.

M es el contador para el array P en la sección de comprobación de los números.

Comentarios

Si desea jugar con sus propios cartones de BINGO y dejar que el ordenador se encargue de cantar los números, entonces tendrá que suprimir los siguientes módulos del programa.

1.3. Prepara los cartones de los jugadores.

1.4. Visualiza los números de los jugadores.

3.1. Idem que el punto 1.4.

Estos dos últimos módulos son la misma rutina. También se puede alterar el programa e incluir, como opción dos maneras de jugar distintas, manual y automática.

```
3990 REM *****************
         *  SUBRUTINA DE  *
         *  COMPROBACION  *
         * DE  RESULTADOS *
         *****************
4000 PRINT
4010 REM *****************
         * SELECCIONA EL *
         *CARTON  GANADOR*
         *****************
4020 PRINT "TECLEA CARTON GANADO
R (Q,R,S,T)"
4030 INPUT A$
4040 IF A$<>"Q" AND A$<>"R" AND
A$<>"S" AND A$<>"T" THEN  GO TO
4020
4050 CLS
4060 DIM V(15)
4070 FOR N=1 TO 15
4080 IF A$="Q" THEN  LET V(N)=Q(
N)
4090 IF A$="R" THEN  LET V(N)=R(
N)
4100 IF A$="S" THEN  LET V(N)=S(
N)
4110 IF A$="T" THEN  LET V(N)=T(
N)
4120 NEXT N
4125 REM ********************
         *COMPROBAR  NUMEROS*
         ********************
4130 LET M=1
4140 FOR N=1 TO 15
4150 IF V(N)=P(M) THEN  PRINT V(
N);" CORRECTO"
4160 IF V(N)=P(M) THEN  GO TO 42
00
4170 LET M=M+1
4180 IF M=Z+1 THEN  GO TO 4220
4190 GO TO 4150
4200 NEXT N
4210 GO TO 4245
4220 PRINT "*CARTON ";A$;" NO ES
TA COMPLETO*"
4230 PRINT V(N);" NO SE HA CANTA
DO"
4235 REM *********************
         *VUELVE AL PRINCIPIO*
         *  PARA  CONTINUAR  *
         *********************
4240 GO TO 4260
4245 PRINT ,,TAB 4;" ** ENHORABU
ENA ** ";TAB 4;"$$$$$$$$$$$$$$$$
$$$"''''"FIN DEL JUEGO, USA 'RUN'
 PARA   EMPEZAR DE NUEVO"
```

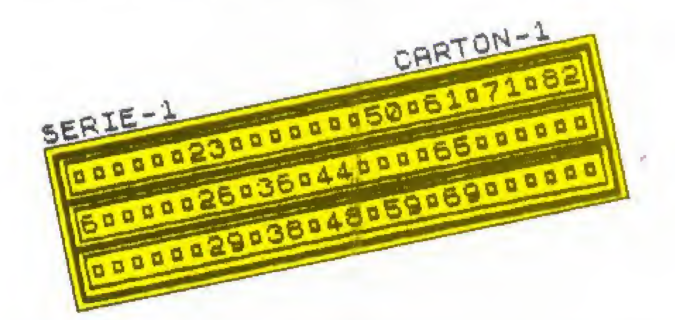
Bingo de Paraninfo Soft
Jugar sin salir de casa

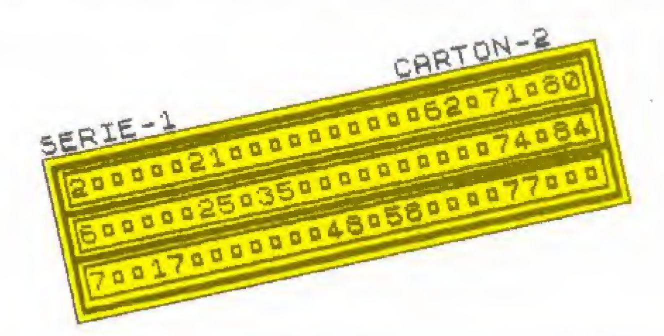
Existen varios programas comerciales para jugar al Bingo. Comentamos brevemente el de **Paraninfo Soft,** cuyo propósito es facilitarnos los cartones y ayudarnos cantando los números y verificando si hemos obtenido línea o bingo.

La cinta viene con dos versiones del juego: en una de las caras se nos permite jugar con los cartones que se incluyen con el *cassette,* con un número máximo de 20. Cada jugador selecciona su cartón, y después le indicamos al programa los números de los cartones seleccionados. En la otra versión es el programa quien nos prepara los cartones aleatoriamente, debiendo cada jugador elegir uno de ellos.

El programa puede hacer copia por impresora de los cartones; para ello debemos usar la ZX printer o impresora compatible, o bien un *Interface* de impresora que soporte el comando COPY.

Existen dos modalidades de juego: el programa puede ir sacando los números automáticamente, con intervalo variable, o esperar a que nosotros pulsemos una tecla para presentarnos el número siguiente. Los números que ya han salido quedan expuestos en un tablero en la parte derecha de la pantalla, mientras el último número





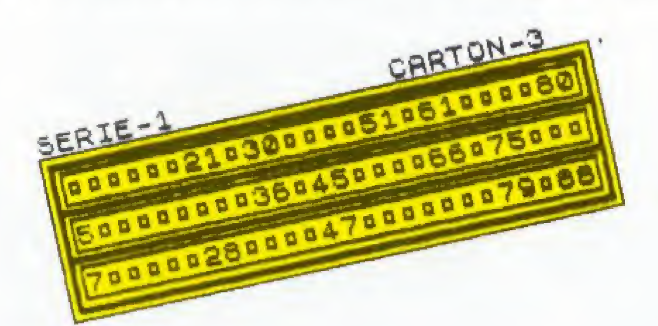

```
4250 REM ************************
         *GO TO FIN DE PROGRAMA*
         ************************
4255 GO TO 1200
4260 RETURN
4270 REM ********************
         * FIN DE SUBRUTINA *
         ********************
4900
5000 REM *********************
         * SUBRUTINA DE PLOT *
         *********************
5020 LET B$(N)=STR$ A(N)
5030 LET C=CODE B$(N)(1 TO 1)
5040 IF LEN B$(N)<2 THEN  GO TO
5060
5050 LET D=CODE B$(N)(2 TO 2)
5060 FOR R=0 TO 7
5070 LET E=PEEK (15360+8*C+R)
5080 LET W=128
5090 FOR G=0 TO 7
5100 IF E<W THEN  GO TO 5150
5110 PRINT AT 2*R+5,2*G+1;"██"
5120 PRINT AT 2*R+6,2*G+1;"██"
5130 LET E=E-W
5140 GO TO 5170
5150 PRINT AT 2*R+5,2*G+1;"  "
5160 PRINT AT 2*R+6,2*G+1;"  "
5170 LET W=W/2
5180 NEXT G
5190 IF LEN B$(N)<2 THEN  GO TO
6020
5200 LET F=PEEK (15360+8*D+R)
5210 LET W=128
5220 FOR H=0 TO 7
5230 IF F<W THEN  GO TO 5280
5240 PRINT AT 2*R+5,2*H+16;"██"
5250 PRINT AT 2*R+6,2*H+16;"██"
5260 LET F=F-W
5270 GO TO 6000
5280 PRINT AT 2*R+5,2*H+16;"  "
5290 PRINT AT 2*R+6,2*H+16;"  "
6000 LET W=W/2
6010 NEXT H
6020 NEXT R
6040 RETURN
6050 REM ********************
         * FIN DE SUBRUTINA *
         ********************
6060
6070 REM ********************
         * FIN DEL PROGRAMA *
         ********************
```

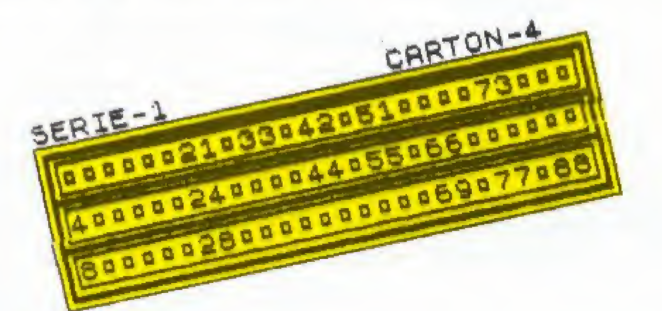
que ha salido se nos presenta ampliado en la parte izquierda de la pantalla.

Otra opción del programa nos deja que sea el programa quien nos avise cuando alguno de los jugadores ha conseguido línea o bingo. Hay que tener cuidado, ya que el programa sólo avisa el cartón de numeración más baja que consiga línea o bingo. Si a la vez hubiera otro, se le debe pedir verificación al programa. Este nos pedirá que introduzcamos los números de nuestro cartón y nos dirá si, efectivamente, hemos compartido el premio. Esta será la única manera de conseguir premio si le dijimos al programa que no cantara línea o bingo.

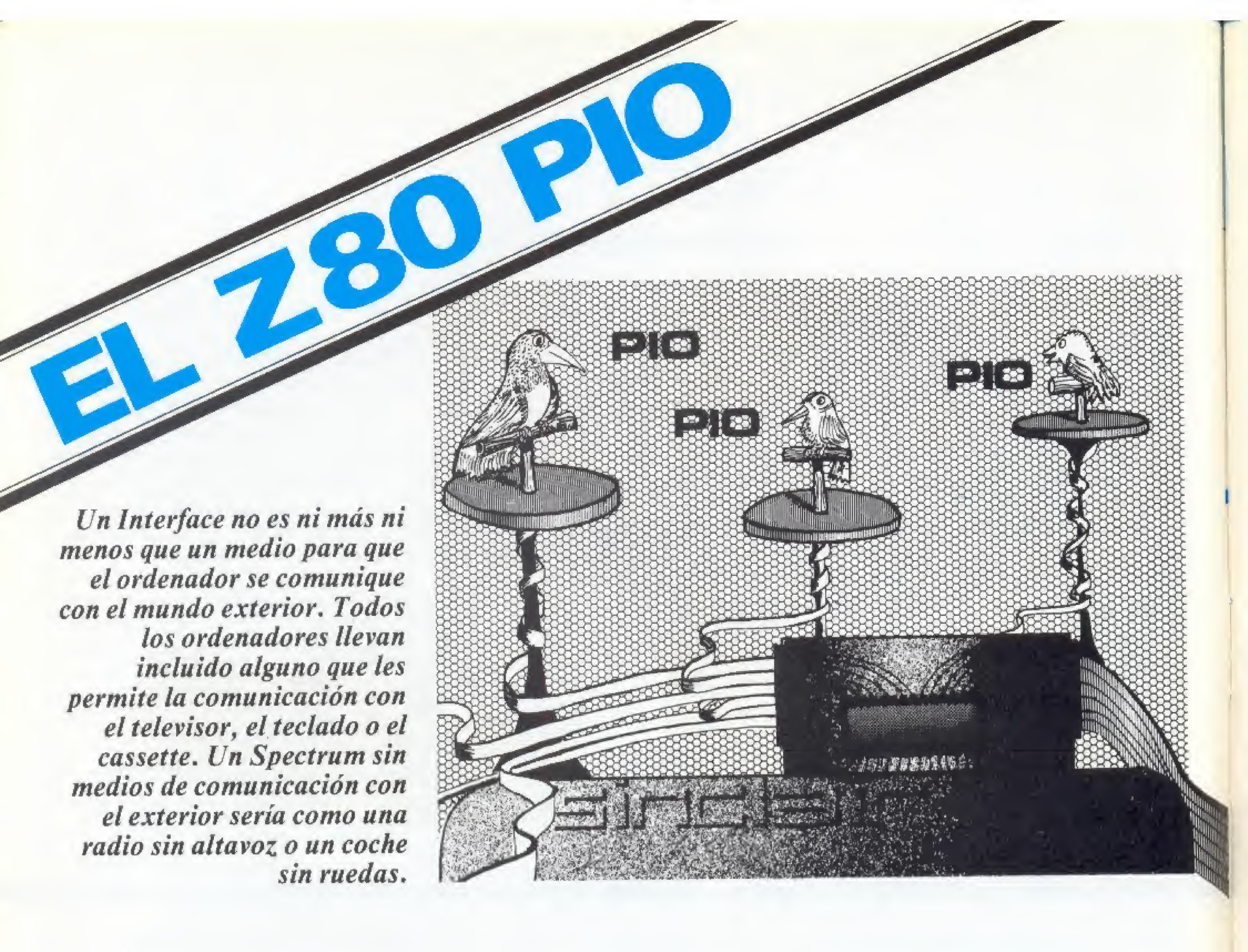
Cantar ¡BINGO! es fácil, lo difícil es hacer trampa al programa.

En suma, nos encontramos con un programa útil para los amantes del Bingo, ya que con él nunca se nos caerán las bolas por el suelo, y tampoco se nos adelantará el listo de turno aprovechando que nos ha tocado a nosotros cantar los números. Será el Spectrum quien se encargará de estas desagradables tareas.





EL Z80 PIO



Un Interface no es ni más ni menos que un medio para que el ordenador se comunique con el mundo exterior. Todos los ordenadores llevan incluido alguno que les permite la comunicación con el televisor, el teclado o el cassette. Un Spectrum sin medios de comunicación con el exterior sería como una radio sin altavoz o un coche sin ruedas.

Parece claro, según esto, que las comunicaciones con el mundo exterior forman parte básica e imprescindible para el manejo y utilización de un sistema tan sofisticado como lo es un Ordenador Personal.

Para el buen entendimiento de un Spectrum con el mundo exterior es imprescindible adaptar la información a los requerimientos del corazón de nuestro Spectrum, el microprocesador Z80-A. Esta es la función de los interfaces.

El Spectrum está dotado de un circuito integrado denominado ULA que es el encargado de las comunicaciones con teclado, pantalla, cassette, altavoz, etc.

¿Por qué no tener nosotros un Interface para las comunicaciones exteriores que nos interesen?

El montaje, aquí presentado, nos permite el acceso al exterior de una manera muy versátil.

Estamos hablando de un integrado de la familia Z80 llamado Z80-PIO.

Z80-PIO

La CPU Z80 tienen dos tipos de instrucciones específicas para la comunicación con los dispositivos de entrada/salida. IN utilizada para introducir un dato desde el exterior y OUT para transmitir un dato al exterior.

Existe un pin (el n.º 20) llamado IORQ en la CPU Z80, que se activará cuando se realice una de estas dos instrucciones. Este pin nos permitirá diferenciar si la CPU está haciendo referencia a una posición de memoria (IORQ = "1") o bien lo está haciendo a una dirección de entrada/salida (IORQ = ∅).

Su versatilidad le permite múltiples aplicaciones

La Z80-PIO (*Peripherial Input Output*) es un dispositivo que contiene dos *ports* de entrada/salida (el A y el B), cada uno de ellos de ocho bits; además, cuenta con cuatro señales de protocolo. En la *figura a* se describen la disposición de pines de este circuito integrado.

Para el *port* A existen cuatro modos de direccionamiento:

— Modo ∅ o modo sólo salida. Sirve para enviar datos al exterior desde la CPU.

— Modo 1 o modo sólo entrada. Sirve para tomar datos desde el exterior y enviarlos a la CPU.

— Modo 2 o modo bidireccional. Engloba los dos anteriores.

— Modo 3 o modo control. Cada línea del *port* tiene un significado diferente y cada una de ellas puede actuar como línea de entra-

da, de salida o de petición de interrupciones.

El *port* B tiene los mismos modos de funcionamiento que el A excepto el modo 2.

Para la utilización de un *port* es imprescindible indicarle el modo de funcionamiento que nos interese. Esto se hará enviando previamente un comando de selección de modo de funcionamiento que más adelante se detallará.

Direccionamiento

En el Spectrum (véase cap. 23 del manual) para las operaciones de entrada/salida con el teclado, pantalla, etc., se utilizan los cinco bits de menor peso del Bus de direcciones para direccionar los *ports* correspondientes a estas comunicaciones.

Para evitar posibles interacciones con los *ports* específicos del Spectrum será conveniente que los bits A0, A1, A2, A3, A4 estén a «1», es decir, que las direcciones asignadas a nuestros montajes contengan en estos cinco bits un «1».

El pin 6 (B/A) de la Z80-PIO es la entrada de selección de PORT. Si éste es «0» nos estaremos refiriendo al *port* A y si es «1» al *port* B.

El pin 5 (C/D) indica, si está a «1», que el contenido del Bus de datos es un comando de selección de modo de funcionamiento, y si está a «0» la información contenida en dicho Bus será un dato.

El pin 4 (CE), activo a nivel bajo, indica a la Z80-PIO que puede utilizar el Bus de datos.

Con motivo de hacer coincidir la dirección asignada al Port A con la utilizada por el KEMPSTON JOYSTICK hemos unido en nuestro montaje el pin B/A de la Z80-P10 con el A5 del Bus de direcciones, el C/D con el A6 y el C/E con el A7, quedando configurada la siguiente tabla de direcciones:

Uso de la PIO

Para la utilización de un *port*, previamente hay que seleccionar su modo de funcionamiento, quedando éste inalterado hasta la próxima selección o un apagado del ordenador.

Selecciones de modo de funcionamiento:

— Modo 0 (sólo salida). Para la selección de este modo de funcionamiento hay que enviar el comando hexadecimal 0F equivalente a 15 en decimal.

En concreto, el *port* A quedará programado en este modo ejecutando desde el Basic la instrucción: OUT 95,15. Para el *port* B sería: OUT 127,15.

se presenten en los hilos del *port* correspondiente.

Por ejemplo: haciendo LET v = IN 31, el dato presente en el *port* A quedará almacenado en la variable v. Con LET v = IN 63, sería el dato presente en el *port* B el almacenado en la variable v.

— Modo 2 (bidireccional). Este modo sólo es posible en el *port* A.

Se selecciona haciendo: OUT 95,143.

En este modo, cuando se haga OUT 31,DATO, el DATO aparecerá, como en el modo 0, en los hilos del *port* A y cuando se haga LET v = IN 31, el funcionamiento será el mismo que el del modo 1.

		A7	A6	A5	A4	A3	A2	A1	AØ	hex.	dec.
PORT A	COMANDO	0	1	0	1	1	1	1	1	5F	95
	DATO	0	0	0	1	1	1	1	1	1F	31
PORT B	COMANDO	0	1	1	1	1	1	1	1	7F	127
	DATO	0	0	1	1	1	1	1	1	3F	63

En este modo de funcionamiento podremos enviar al exterior por el *port* en cuestión, cualquier dato, comprendido entre 0 y 255, que aparecerá en binario en los ocho hilos del *port*. De esta manera, para el *port* A siempre que hagamos OUT 31,DATO aparecerá el DATO, escrito en binario, en los hilos A0 a A7 del *port* A de la PIO. Para el *port* B sería: OUT 63,DATO y el DATO aparecerá en los hilos B0 a B7 del *port* B de la PIO.

— Modo 1 (sólo entrada). Para seleccionar este modo de funcionamiento habrá que hacer:

Port A: OUT 95,79

Port B: OUT 127,79

Este modo permite la lectura de los datos, escritos en binario, que

— Modo 3 (control). Se selecciona haciendo:

Por A: OUT 95,207

Port B: OUT 127,207

En este modo de funcionamiento se puede escoger de cada *port* los hilos que deseemos como entradas o como salidas. Para indicar qué hilos serán de entrada y cuáles de salida hay que enviar un segundo comando cuyo valor se obtendrá considerando el siguiente formato:

Para los hilos que se deseen como entrada hay que poner un «1» y para los de salida se pondrá un «0».

Haremos entonces:

Port A: OUT 95, BIN $X_7X_6X_5X_4$ $X_3X_2X_1X_0$

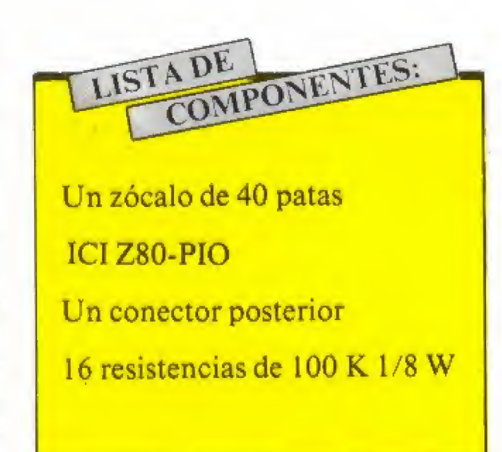
Port B: OUT 127,BIN $X_7X_6X_5X_4$ $X_3X_1X_0$.

Por ejemplo: Si queremos tener en el *port* A los hilos A4, A3 y A1 como entradas y el resto como salidas tendríamos que hacer:

OUT 95,207: OUT 95,BIN 00011010

Selecciona modo 3: Especifica líneas I/O.

Haciendo LET v = IN 31 se guardará en la variable v el dato compuesto por la información presente en las líneas definidas como



A7	A6	A5	A4	A3	A2	A1	AØ
B7	B6	B5	B4	B3	B2	B1	BØ
X_7	X_6	X_5	X_4	X_3	X_2	X_1	$X_Ø$

entradas y la última información enviada por líneas de salida.

Haciendo OUT 31, DATO sólo enviaremos la información correspondiente a las líneas seleccionadas como salidas ignorando la correspondiente a las líneas de entrada.

Montaje

Este montaje presenta cierta dificultad por el hecho de estar realizado en circuito impreso de doble cara junto con la necesidad de tener que hacer bastantes soldaduras en ambas caras.

Los taladros señalados con la letra p, en la figura adjunta, son puentes de conexión entre las dos caras. Estos puentes se pueden realizar, por ejemplo, con trozos de patillas de resistencias.

Es indispensable el uso de un zócalo de 40 pines para la PIO con el fin de evitar la inutilización del integrado por excesivo calentamiento al efectuar las soldaduras.

En el prototipo se ha soldado directamente el conector a la placa de circuito impreso, aunque pueden soldarse a través de unos hilillos de los usados normalmente para cableado.

Hay que recordar que nuestro *port* A (cuya dirección es la 31) coincide con la utilizada por el KEMPSTON JOYSTICK lo que nos permite utilizar el montaje descrito como *Interface* para dicho tipo de *Joystick*. Para ello habrá que hacer funcionar el *port* A en modo de sólo entrada.

El KEMPSTON JOYSTICK utiliza la siguiente asignación de bits:

A4: Disparo.

A3: Movimiento hacia arriba.

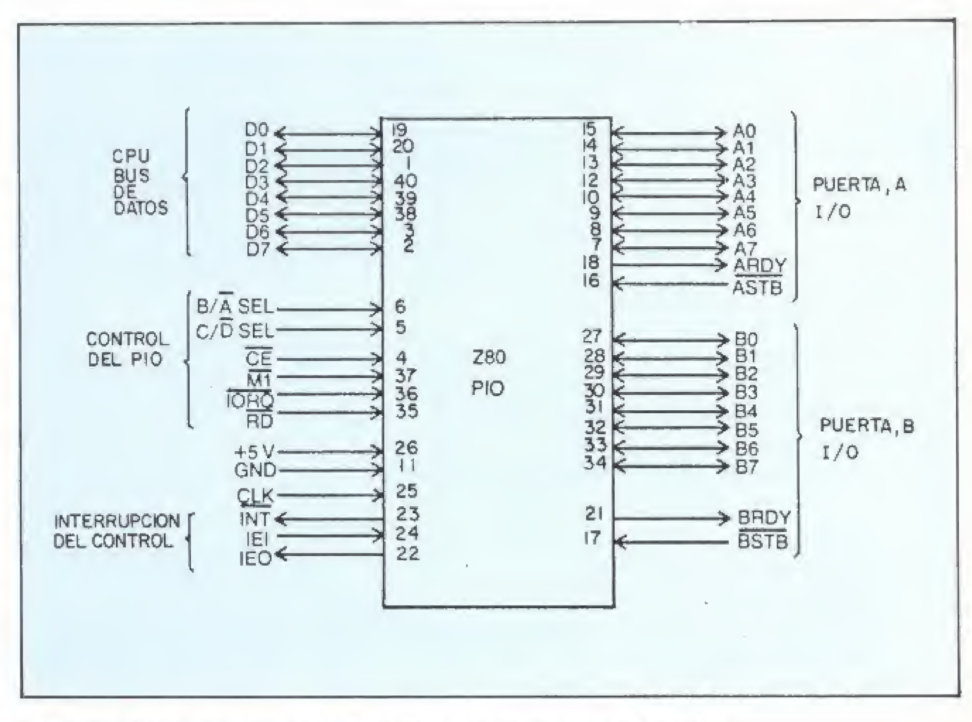
A2: Movimiento hacia abajo.

A1: Movimiento hacia la izquierda.

A0: Movimiento hacia la derecha.

Estas señales son activas a nivel alto, es decir, se efectuará el movimiento correspondiente cuando haya tensión de 5 v en el bit asignado. Los restantes bits, no mencionados (A7, A6, A5), no están conectados.

Nosotros hemos utilizado este montaje con un *Joystick* SPECTRAVIDEO mod. Quick Shot



Esquema de las funciones de la Z80-PIO. Los números en el interior corresponden a las asignaciones de las patas del circuito integrado.

TM II y hemos conectado los hilos según la relación:

A0: marrón.

A1: verde.

A2: azul.

A3: blanco.

A4: naranja.

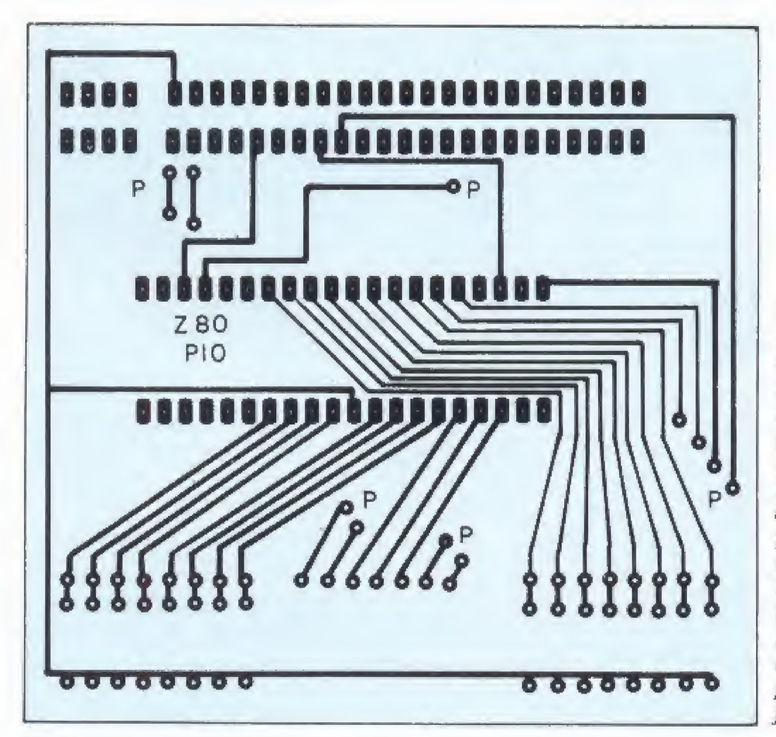
A5: rojo.

Una particularidad muy importante, para el buen funcionamiento del *Joystick*, es la de que hay que hacer, antes de cargar el juego deseado, la instrucción: OUT 95,79, con el fin de que el *port* A quede debidamente programado en el modo de sólo entrada.

Futuros montajes con la PIO

Evidentemente, la utilización de la Z80-PIO como *Interface* para un *Joystick* sería un derroche de trabajo y dinero.

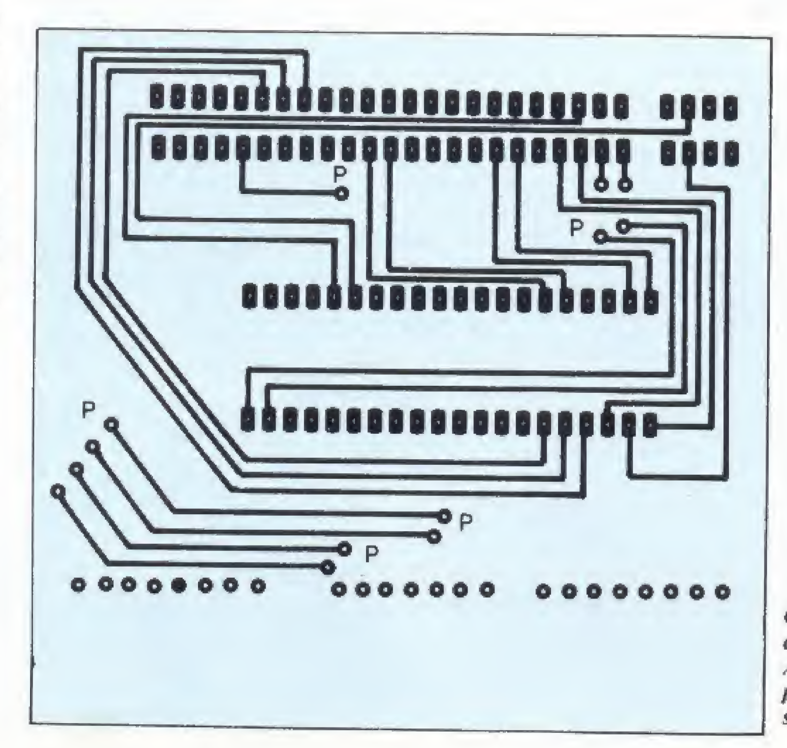
Este montaje, al que podríamos bautizar como *Interface ∅*, tiene infinidad de aplicaciones (sólo limitadas por la imaginación), tales como: Controlador Doméstico, Voltímetro de baja frecuencia, Osciloscopio de baja frecuencia con memoria, Cuentavueltas múltiple para Scalextric y similares, Controlador de tráfico para trenes eléctricos, Generador programable de baja frecuencia, Control de invernaderos, peceras, etc., *Interfaces* serie y paralelo para impresoras, etc.



Cara superior de la placa de circuito impreso. Los terminales marcados P corresponden a puentes entre la cara superior y la inferior. En la parte superior de la figura están las pistas del conector al bus de expansión del Spectrum. Las resistencias deben soldarse en las salidas de los dos ports en la parte inferior de la figura.

Esperamos seguir con algunas aplicaciones de este montaje en nuestros próximos números. Y también esperamos que nos enviéis los desarrollos que realicéis a partir de este montaje ampliando las posibilidades de comunicación del Spectrum.

Leandro Rodríguez

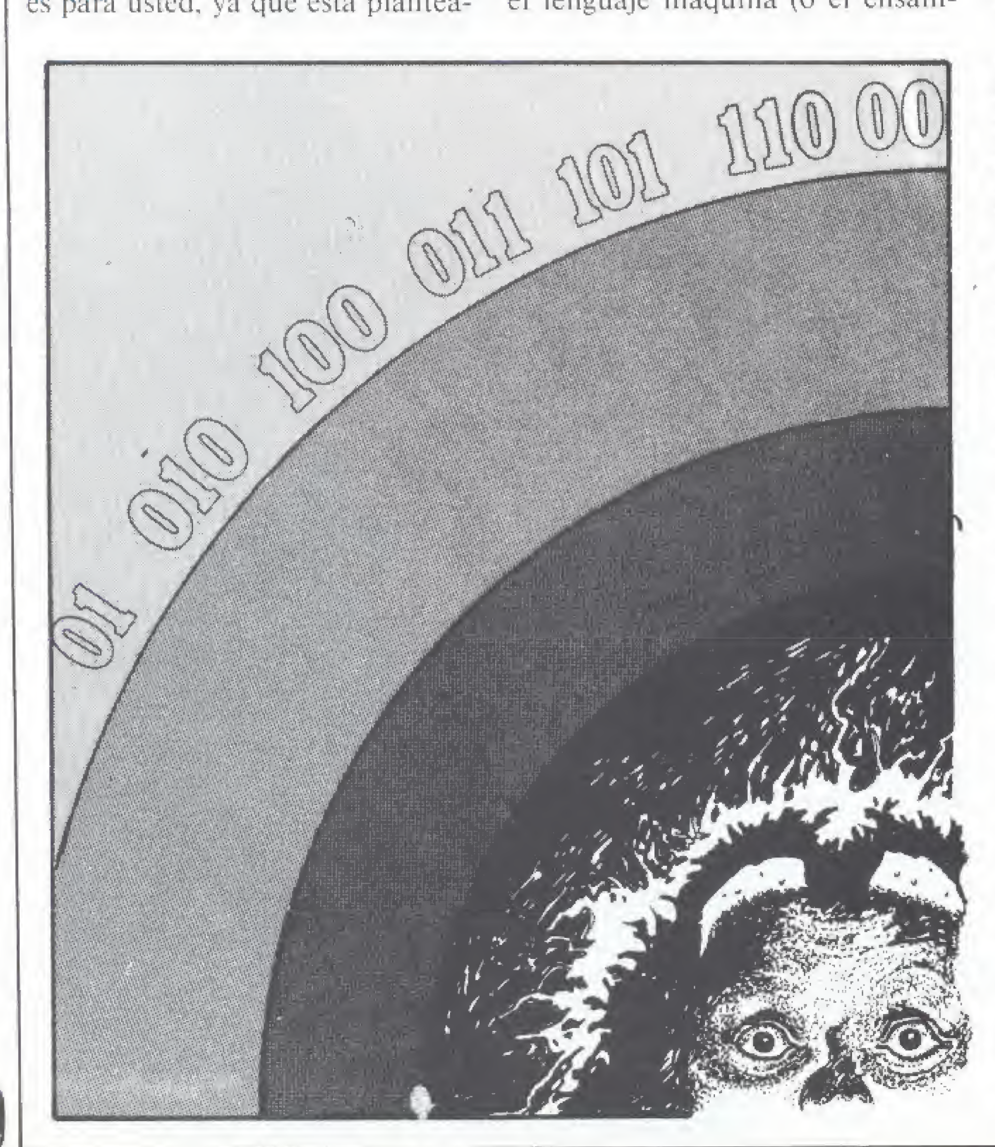


Cara inferior del circuito impreso. Atención a los puentes con la cara superior.



En este número empieza una serie dedicada a explicar el funcionamiento y utilidad de ese lenguaje tan misterioso del que muchos hablan pero pocos parecen entender realmente.

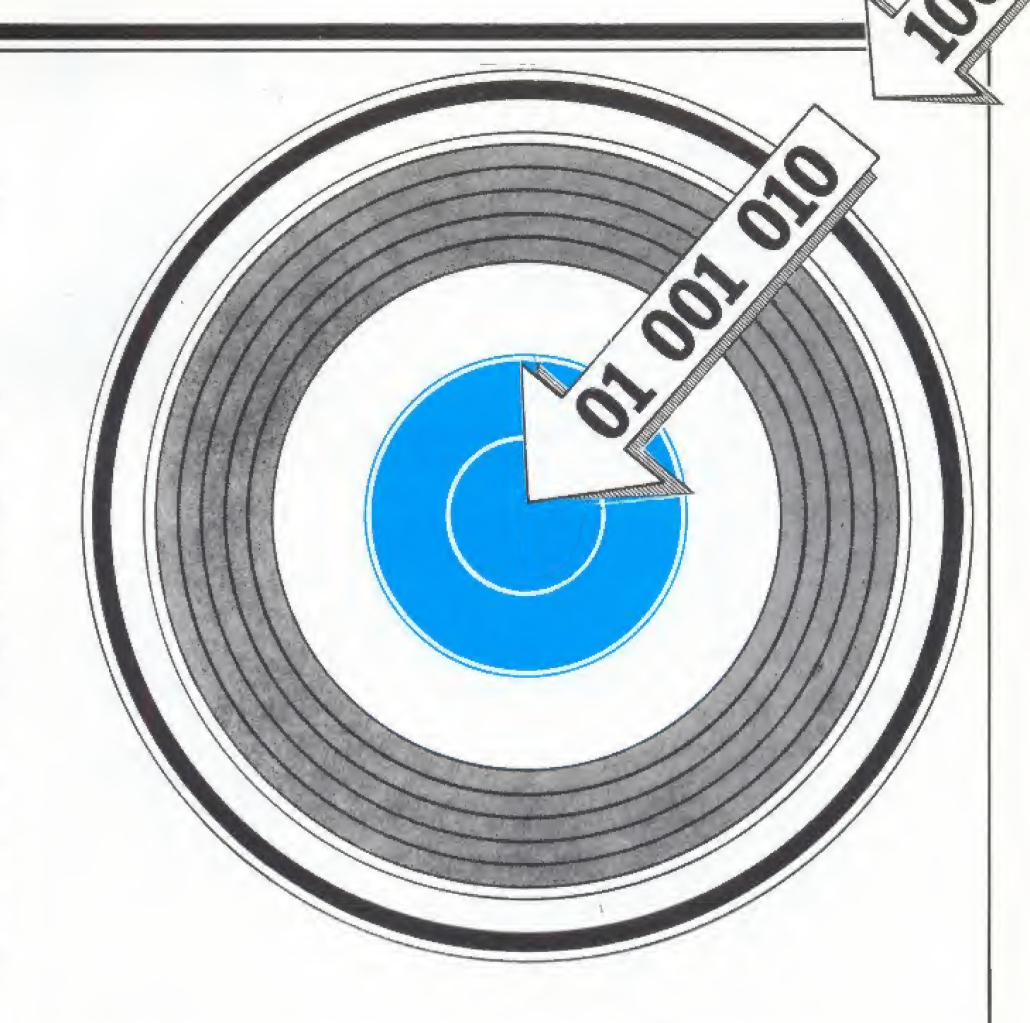
Si usted ya maneja el lenguaje máquina y sabe el significado de términos tales como «LDA», «IX», *«byte»* y *«bit»*, entonces esta serie NO es para usted, ya que está planteada como un aprendizaje desde los principios básicos subiendo poco a poco hasta dominar este «idioma» de entendidos. En definitiva, va destinada a aquéllos que creen que el lenguaje máquina (o el ensamblador) son los cuchicheos que se cuenta un ordenador a otro, o la persona que los monta en la fábrica o para los que creen que para manejarlo se necesita soldador y estaño (cosa totalmente falsa).



Quizás en algunas partes, como sucede este mismo mes, nos extendamos mucho en explicaciones teóricas que a alguien le podrán parecer superfluas y destinadas a llenar hojas, pero a los escritores técnicos (por lo menos a nosotros) no les pagan por palabra escrita, y si algo se incluye es porque será necesario más adelante para comprender otras instrucciones más prácticas.

Junto a la serie en sí misma irán apareciendo, a lo largo de los meses unos comentarios sobre los programas ensambladores existentes en el mercado. Estos programas, cuya función explicamos más adelante, se pueden considerar indispensables para trabajar en lenguaje máquina debido al trabajo que ahorran y si no posee ninguno piense en comprarlo si quiere participar activamente en este «serial»; para elegir el que más le convenga nada mejor que leer estos comentarios.

Por último, debemos hacer una recomendación importante. En informática, al igual que en otras ciencias, el saber la teoría no implica necesariamente el saber hacerlo prácticamente y se puede llevar desagradables sorpresas si se pone a hacer un gran programa sin



haber practicado previamente parte por parte. Si tiene un ordenador es para usarlo, no lo olvide.

Cómo funciona un ordenador

Está demostrado científicamente que los ordenadores *no* funcionan porque tengan dentro una pandilla de enanitos que manejen ábacos a mucha velocidad, ni tampoco por intuición, todos poseen una estructura básica muy similar que, aunque no necesaria para programas, sí es conveniente saber cómo es para poder ver el sentido lógico de las instrucciones.

La estructura interna de un ordenador es la que se ve en la figura 1 en la que se distinguen varios bloques tales como microprocesador, memoria (RAM y ROM) y dispositivos de entrada y salida (E/S). Todos ellos son comunes a todos los ordenadores y las diferencias residen en el tipo de estos dispositivos, pero no en su existencia, ya que siempre están.

El más importante de todos estos bloques (o por lo menos más famoso), es el microprocesador. Este es lo que se denomina «corazón» de la máquina, aunque más bien se le puede denominar el cerebro ya que es el que se encarga de realizar todos los procesos y de dirigir a los demás. En el caso del Spectrum, el microprocesador se denomina Z-80 y es ampliamente conocido ya que lo llevan otros ordenadores existentes en el mercado (Amstrad, MSX y todos los que funcionan con CP/M). Este viene a ser el cartero que reparte las cartas mientras que los demás constituyen los buzones o los almacenes de las carterías.

La memoria (RAM o ROM) se asemeja a los cajetines existentes en las oficinas de correos en los que se distribuyen las cartas para ser repartidas luego. Aunque como veremos estos cajetines tienen unas características bastante especiales que los diferencian de los reales.

Por último los dispositivos de entrada y salida constituyen los buzones donde dejamos las cartas y de donde las recogemos, es decir, por donde entra y sale la información. Los ejemplos más claros de estos dispositivos son el teclado y la pantalla, aunque también pueden existir otros como son la impresora, los *joystick*, etc.

Los sistemas de numeración. O cómo 1000 puede ser distinto de 1000

Esta afirmación tan sorprendente tiene su razón de ser. Nosotros los humanos estamos acostumbrados a operar en decimal, lo que quiere decir, ni más ni menos, que 9 más 1 es 10. Pero los ordenadores tienen la costumbre de operar en binario. En este caso resulta sorprendente ver que 1 más 1 es 10, aunque muchas veces (por ejemplo, trabajando en BASIC) se nos muestran los números en decimal para simplificarnos la tarea. En lenguaje máquina esto no sucede y hay que acostumbrarse a operar en binario y en hexadecimal, que es otro modo de expresar los números.

Es muy probable que con el párrafo anterior se haya armado un lío, y no es extraño, ya que éste es uno de los puntos más difíciles de explicar de todos. Para simplificar todo y hacerlo más comprensible, vamos a basarnos no en números, sino en cantidades, para ello coja una caja de cerillas (lo sentimos por aquéllos que usan mechero) y dispóngalas a su lado. Nosotros las representaremos aquí por asteriscos (*) pero el proceso será el mismo.

Ya sabemos que en decimal «*» es 1, «**» es 2 y así sucesivamente hasta «*********» que significa 9. También sabemos que si juntamos «*» (1) y «*********» (9) obtenemos «**********» (10). Esto lo ex-



presamos diciendo que «9 más 1 es 10». Esto lo comprendemos todo y sabemos que 90 más 10 es 100 (perdone que no lo pongamos con asteriscos pero no tenemos espacio) ya que 0 más 0 es 0 y 1 más 9 es 10. Pero en binario es distinto, ya que, sólo existen los números cero y uno, en este sistema cero significa que no tenemos ninguna cerilla y uno es «*». Pero uno más uno (¡¡¡Preste atención!!!) es 10, lo que haciéndolo con cerillas es «*» más «*» es igual a «**». Por lo que, por la relación que acabamos de expresar significa que 10 es «**» en binario, mientras que en decimal esta cantidad se expresa como 2. Por otra parte, en binario sigue siendo cierto que 0 más 1 es 1, por lo que si en binario decimos 10 más 1, queda 11, que es «**» más «*», o sea «***», y 11 más 1 (también en binario) es 100 (1 más 1 es 10 y nos llevaremos 1, en el siguiente sucede lo mismo y queda el 100) que con cerillas da «****». Vemos por tanto que los números son relativos y que pueden expresar diversas cantidades dependiendo de la base en que estemos trabajando. Para no confundirnos a partir de ahora pondremos un «b» detrás del número si éste está en binario y si no hay nada es que está en decimal.

Al igual que hemos visto el sistema binario y el decimal, existe una cantidad infinita de sistemas de numeración, ya que la única diferencia entre unos y otros está en el significado del término, «10», o lo que es lo mismo, la cantidad máxima que se puede expresar con una sola cifra, que será una menos que la indicada por «10». En el sistema decimal (decimal = base 10) un «10» es «**********», en el binario (binario = base 2) «10» es «**». De lo que se deduce que en cada sistema el «10» es la cantidad indicada por la base y sólo existirán números de un solo dígito inferiores a éste, ya que si no, habría dos símbolos para expresar la misma cantidad (por ejemplo 10 y 2 en binario) lo que va en contra de los fundamentos de las matemáticas, naturalmente es necesario saber en qué base se expresa la base de un sistema (si nos dicen que operamos en decimal, o sea en base 10, hay que saber en qué base está este 10) pero ésta, por definición siempre es la usada por los humanos (decimal = «**********» y binario = «**»).

Si por ejemplo, nos dicen que trabajamos en base cinco, sabremos que en ese sistema, «10» es «*****» y, por tanto 4 + 1 = 10. Asimismo, sólo existirán en base los dígitos «0, 1, 2, 3 y 4» y no tendrán sentido números como «517» o «912» ya que incluyen cifras no válidas. Si decimos que operamos en base octal (8) el proceso será el mismo, pero con la excepción de que sólo existen «0, 1, 2, 3, 4, 5, 6 y 7» y 10 = «********».

En todos los sistemas las operaciones se hacen como sabemos, pero teniendo en cuenta el límite de dígitos. Tomemos como ejemplo la suma y la resta (las más usadas), si trabajando en base octal, nos ponen el siguiente problema: 451 + 743, la solución se hallará diciendo: 3 y 1, 4 y 5 NO es 9 (este número no existe), si no que debemos expresar el 5 como 4 + 1 y queda 4 + 4 + 1, sabiendo que 4 + 4 es 10, queda que 4 + 5 es 11, en la tercera columna queda 4 más 7 más 1 que nos llevamos de antes, ahora bien, 7 más 1 es 10 y diez más cuatro es 14, por lo que la suma nos da de resultado 1414 (en octal). En los sistemas de numeración que hemos estudiado hasta ahora, los procesos son más sencillos, ya que en decimal ya sabemos (el que hemos manejado siempre) y en binario sólo existen tres operaciones distintas 0 + 0 = 0, 0 + 1 = 1 (ambas similares a las de decimal) y 1 + 1 = 10. Sabiendo esto no resulta ningún problema operar.

Al principio del capítulo comentamos otra base que es la denominada hexadecimal. Como hexadecimal significa 16, sabremos que en esta base 10 = «****************», o lo que es lo mismo, 16 en decimal. Pero nos encontramos con un problema, si 10 tiene este valor, ¿cómo se expresa 9 + 1? Como hemos comentado anteriormente, debe ser un número de una sola cifra ya que no puede haber ningún valor inferior a «10» con dos cifras, pero nuestro sistema de numeración no dispone de lo necesarios, por lo que tendre-

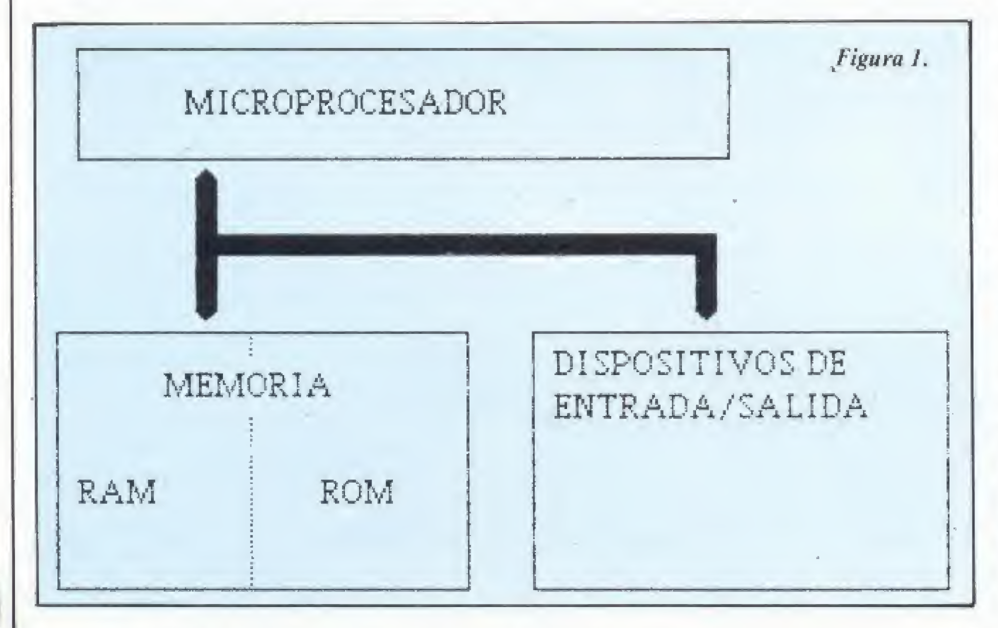
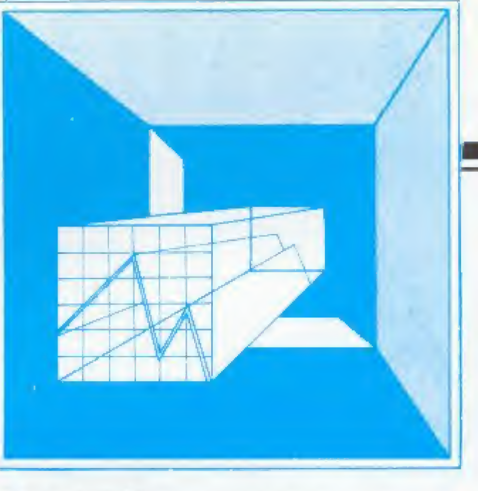


Figura 1.



mos que recurrir a otros. En lugar de inventarlos nuevos recurriremos a un grupo bien conocido: el alfabeto. Sabemos que tenemos que usar seis nuevos (9 + 1, 9 + 2, 9 + 3, 9 + 4, 9 + 5 y 9 + 6), ya que 9 + 7 es «10». Para ello elegiremos las seis primeras letras y se las asignaremos en el mismo orden, por lo que en hexadecimal: 9 + 1 = A, 9 + 2 = B, 9 + 3 = C, 9 + 4 = D, 9 + 5 = E, 9 + 6 = F. Expresado en cantidades sale A = «**********», F = «***************». Evidentemente F + 1 = 10, A + 5 = F, etc. Para representar un número hexadecimal le añadiremos detrás la letra «h» para distinguirlo de los demás.

Un último detalle a explicar es el de pasar de una base a otra. Para pasar de cualquiera a decimal, el proceso es muy sencillo, si nos fijamos, por ejemplo en el sistema binario, en el valor de 1b, veremos que es 1 «*», 10b es 2 «**», 100b es 4 «****». Y 1 es 2 elevado a cero, 2 es 2 elevado a uno, 4 es 2 elevado a dos. Por tanto, si numeramos las distintas posiciones de los números binarios de derecha a izquierda empezando desde cero (figura 2), veremos que el valor en decimal de cada posición es la correspondiente potencia de dos, por lo que para hallar el valor decimal basta multiplicar el correspondiente dígito por su potencia de dos y luego sumarlos todos. Este proceso se muestra en la figura 3.

En hexadecimal el proceso es similar pero los números que debemos multiplicar son la correspondiente potencia de 16 (que es la base en este caso) por el valor correspondiente en decimal del dígito del número (Ah = 10, Bh = 11,... Fh = 15). El proceso para números hexadecimales se muestra en la figura 4.

Con todo esto esperamos que hayan comprendido cómo funcionan las distintas bases y en caso de duda recurran a la caja de cerillas.

Figura 2.

POSICIONES:	7	6	5	4	3	2	1	0
VALORES:	128	64	32	16	8	4	2	1
NUMEROS:								

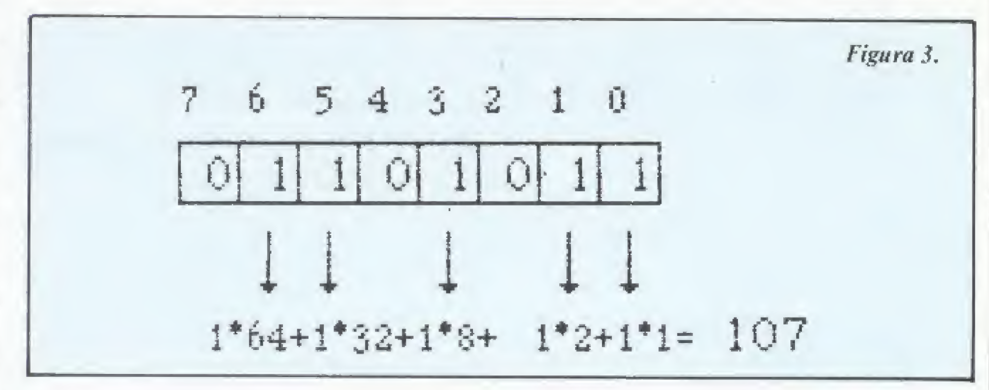


Figura 3.

Algunos términos de uso muy común

Un último punto que se debe tocar antes de pasar a una descripción completa del ordenador, es el de los términos comunes empleados a lo largo de toda la serie. Muchos de ellos son ampliamente manejados por todos aún sin saber qué significan, otros son totalmente desconocidos. De todos ellos haremos una descripción para que se puedan manejar con conocimiento de causa.

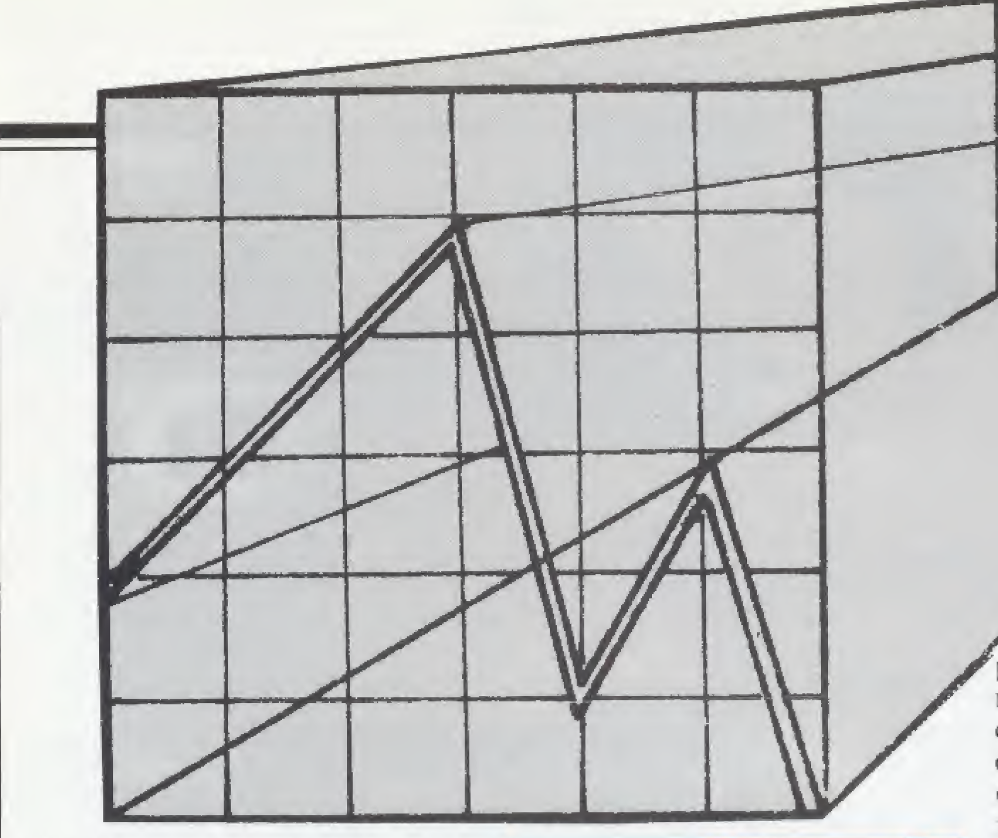
El primero de todos es *«bit»* (no confundir con *byte*). Probablemente habrá oído hablar mucho de él. Este es una especie de interruptor que sólo puede tener dos estados: encendido y apagado. Cuando está encendido decimos que vale 1 y cuando está apagado decimos que tiene un 0. Si ponemos dos *«bits»* juntos, veremos que puede adoptar los siguientes valores entre los dos: «00», «01», «10» y «11». Salta rápidamente a la vista que éste es el sistema binario, por lo que con bits podremos expresar números en sistema binario (y si es necesario luego podremos pasar a otro sistema).

Byte es el segundo término, y sin duda el más famoso, un *byte* es simplemente un grupo de 8 bits, de modo que el valor menor que puede adoptar es 00000000b y el mayor 11111111b, que se corresponden al 0 y al 255 decimales.

«K» es la abreviatura de *Kilobytes,* que No significa 1.000 *bytes,* sino 1.024 *bytes.* La razón de esta diferencia está en que 1.024 es una potencia de 2 y 1.000 no lo es, por lo que al ordenador le resulta mucho más fácil manejar el primer número.

ASCII es la abreviatura de *«American Standard Code for Information Interchange»* y es un sistema de códigos empleados para poder almacenar letras y otros símbolos dentro del ordenador, ya que en principio éste sólo puede almacenar números en sus memorias.

CPU es la abreviatura de *«Central Proccesing Unit»,* en palabras llanas este es el microprocesador



(el Z-80) pero llamado de otra forma.

Nibble, este término bastante poco conocido indica la mitad de un *byte*, es decir, cuatro *bits*.

Software es el nombre que se utiliza para designar cualquier tipo de programa, incluyendo los que vienen metidos dentro de él desde la fábrica. El *software* es pura información (datos o instrucciones para ejecutar programas) y se debe transmitir por algún medio (listado de impresora, cinta de *cassette*, *microdrive*, etc.) pero estos elementos no constituyen el *software*, sino la información que transmiten.

Hardware es el opuesto del anterior, en general es todo componente físico del ordenador, cualquier cosa que se pueda tocar. Normalmente se refiere a los circuitos que lleva. La palabra significaba originalmente en inglés «chatarra».

El término bug surgió cuando, en los primeros ordenadores, las máquinas resultaban averiadas al introducirse polillas entre las válvulas.

Bug es literalmente «bichito» (véase cucaracha, polilla, etc.) pero ha evolucionado por caminos extraños hasta convertirse en un término empleado para indicar un error en un programa. Dice la leyenda que el término se empezó a usar después de que una polilla estropease uno de los primeros ordenadores IBM existentes allá por los lejanos años 60.

El término *«dump»* se utiliza para indicar un volcado de la memoria, como esta frase les habrá dicho lo mismo que la primera (es decir, nada), la explicaremos. Muchas veces cuando falla un programa en lenguaje máquina es necesario mirar todos los contenidos de memoria y ver dónde ha fallado el programa. Para ello se obtiene un listado donde se muestra el contenido de todas las memorias. Este listado es el denominado «volcado de memoria» o «*dump* de memoria».

La pila o *«stack»* es un sistema de almacenamiento de datos muy usado que funciona como una memoria LIFO (ver definición siguiente).

LIFO. Sistema de almacenamiento *«Last In, First Out»*, que significa último en entrar, primero en salir. Y significa que si en un sistema de éstos introducimos los números 7, 45 y 12 por este orden, el primero que podemos sacar es el 12, el siguiente el 45 y por último el 7. Funciona de un modo similar a una pila de platos de la que sólo podemos coger el superior.

FIFO. Abreviatura de *«First In, First Out»* (primero en entrar, primero en salir). Al contrario que el anterior, en este sistema se saca primero el primer número que entró, de modo que si introducimos los mismos datos que en el caso anterior, primero sacaremos el 7, luego el 45 y por último el 12.

Con esto quedan explicados los principales términos. Los demás que necesitemos los iremos explicando según se necesiten.

Fernando García

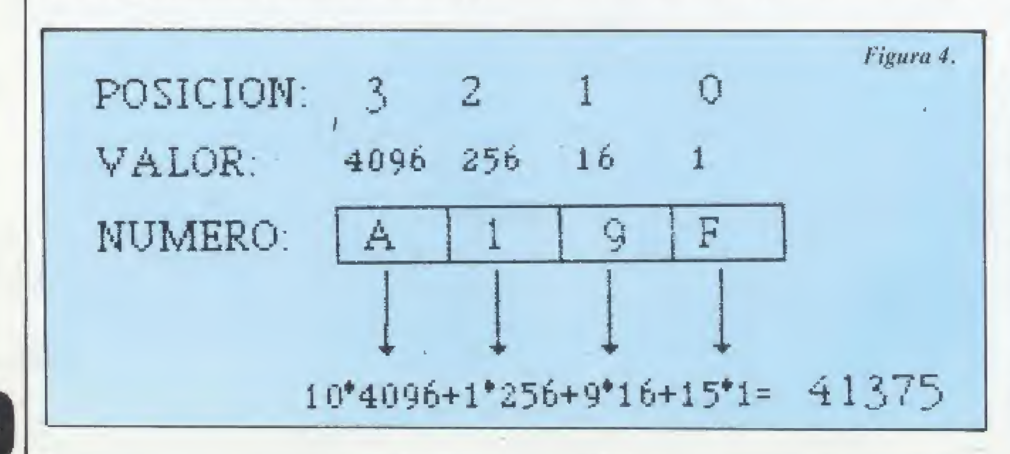
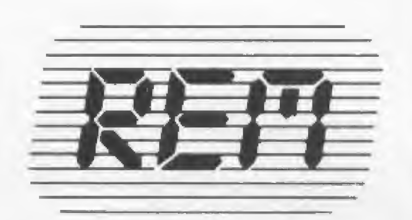


Figura 4.

Guía del comprador de Todospectrum



• Ordenadores personales Hard y Soft.

• Cursos de Basic.

Oficina **RENOVACION EN MARCHA, S. A.**
C/ Espronceda, 34. 28003-MADRID
Tfno. (91) 441 24 78

REMSHOP 1
Galileo, 4. 28015 MADRID
Tfno. (91) 445 28 08

REMSHOP 2
C/ Dr. Castelo, 14. 28008 MADRID
Tfno. (91) 274 98 43

REMSHOP 3
C/ Modesto Lafuente, 33. 28003 MADRID
Tfno. (91) 233 83 19

REMSHOP BARCELONA
C/Muntaner 55 - 0804 BARCELONA
Tfno (93) 253 26 18

REMSHOP LAS PALMAS
C/ General Mas de Gamindez. 45. LAS PALMAS
Tfno. (928) 23 02 90

REMSHOP BILBAO
C/ General Concha, 12 - 48008 BILBAO
Tfno. (94) 444 68 68

REMSHOP OVIEDO
C/ Matemático Pedrayes, 6 - 33005 OVIEDO
Tfno. (985) 25 25 95

ELECTRONICA
SANDOVAL S.A.

DISTRIBUIDORES DE:

COMMODORE-64
ORIC-ATMOS
ZX SPECTRUM
SINCLAIR ZX 81
ROCKWELL'-AIM-65
DRAGON-32
NEW BRAIN
DRAGON-64
CASIO FP-200

ELECTRONICA SANDOVAL, S. A.
C/ SANDOVAL, 3, 4, 6. 28010-MADRID
Teléfonos: 445 75 58 - 445 76 00 - 445 18 70
447 42 01
C/ SANDOVAL, 4 y 6
Centralita 445 18 33 (8 líneas)

CAMAFEO INC.



CASSETTES
DE CALIDAD PROBADA
PARA ORDENADORES

Cada uno	Caja de 10	Caja de 30
C-5 199 ptas.	1.393 ptas.	3.582 ptas.
C-10 209 ptas.	1.463 ptas.	3.762 ptas.
C-15 219 ptas.	1.533 ptas.	3.942 ptas
C-20 229 ptas.	1.602 ptas	4.122 ptas.

Libre de gastos de envio contra reembolso correos

CAMAFEO INC. Dep. 03
José Lázaro Galdiano, 1. 28036 Madrid.

K-BITS

- **ORDENADORES PERSONALES**
- **GESTION**
- **APLICACIONES ARQUITECTURA**
- GARANTIA OFICIAL
- FACILIDADES DE PAGO
- ENVIOS A PROVINCIAS

Barquillo 15 - Tfno.: 232 57 37 - 28004 MADRID

CURSO DE CONTABILIDAD
PARA P y M EMPRESAS

EN ZX SPECTRUM

- Libros Oficiales Contabilidad
- Diarios, Inventarios, Balances, etc.
- Plan General Contable

CENTRO DE ESTUDIOS: SUMAAS
c/. Desengaño, 12 - 3.°-3 28004 Madrid
Telfs.: 221 31 49 - 221 38 35

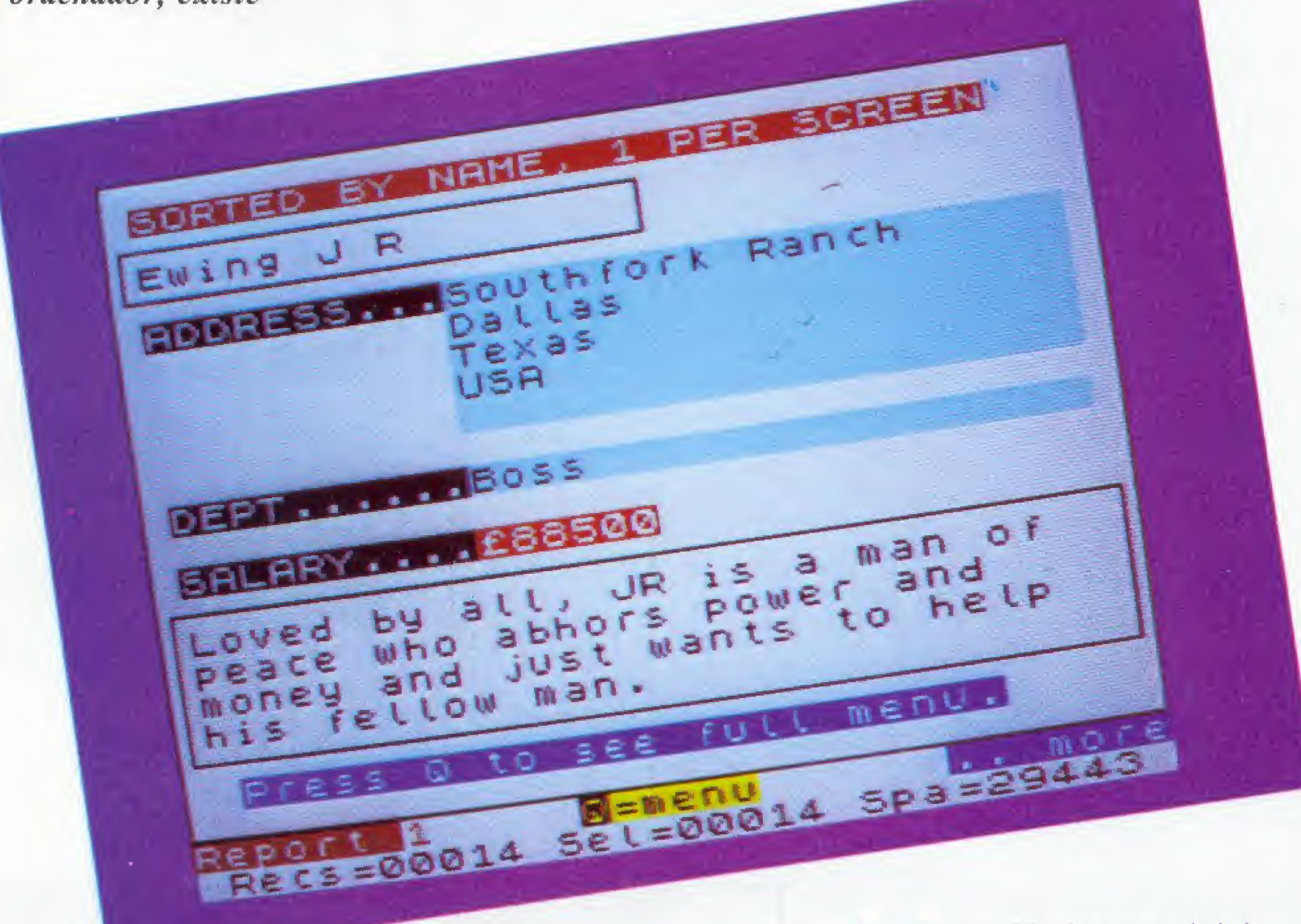
ANUNCIESE por MODULOS

MADRID (91) 733 96 62

BARCELONA (93) 301 47 00

MASTERF

La gestión, actualización y presentación de grandes cantidades de datos es una de las aplicaciones más importantes de los ordenadores personales. A pesar de que el cassette no es el medio de almacenamiento idóneo, debido a su baja velocidad de transferencia y a la imposibilidad de rebobinar o avanzar sin intervención del ordenador, existe



un creciente número de usuarios que no se resigna a dejar de utilizar su Spectrum para la gestión de pequeños ficheros. Para ello existen dos posibilidades: utilizar programas específicos, diseñados para la aplicación concreta que se quiera realizar, o bien un programa de base de datos, mucho más general, pero que no podrá procesar todos los aspectos que deseemos de nuestro fichero, o nos obligará a hacerlo manualmente.

MasterFile intenta cubrir lo mejor permitiéndonos añadir un programa en BASIC que se ejecutará cada vez que se procese un registro. Por este sistema se puede calcular algún campo de cada ficha en función de los contenidos de otros campos, o realizar cuentas del número de ocurrencias de una campo determinado, etc. A esto se le añade una base de datos rápida, con una gran variedad de formatos de escritura.

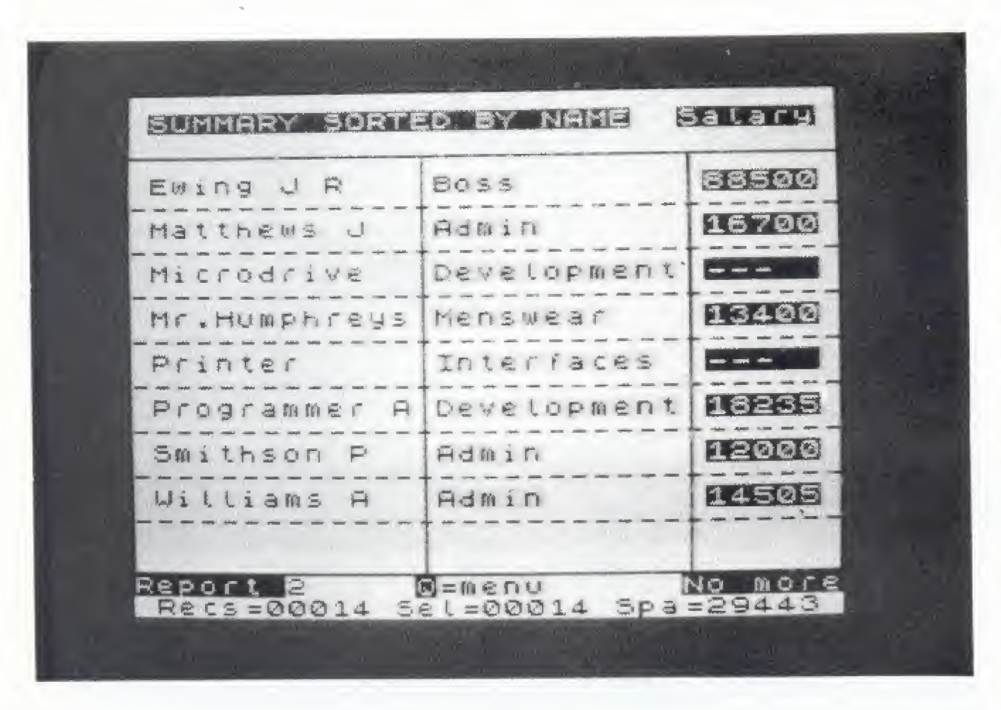
ILE: Base de Datos para Spectrum

Comenzamos, para quienes no estén habituados, por la terminología de la base de datos. Un fichero está formado por un número indeterminado de fichas o registros, cada uno de los cuales se compone de distintas líneas de información (campos). En MasterFile el número máximo de campos es de 26, cada uno de los cuales puede contener hasta 128 caracteres de información. La estructura de cada registro no es rígida, pudiendo llevar los campos en cualquier orden y omitir alguno de ellos. La capacidad de memoria llega hasta 32 K para datos, dependiendo de la cantidad de memoria que utilicemos en nuestro programa.

Una de las características más interesantes en MasterFile es que puede mostrar la información con hasta 36 formatos diferentes, que quedan almacenados en el ordenador. Así, un formato podría listar a los empleados de una empresa por orden alfabético, con nombre, dirección, teléfono y cargo, mientras otro nos proporciona la lista por departamentos, con el cargo y el salario. El cambio de uno a otro es muy sencillo, y no exige más que introducir el número del formato.

La presentación en pantalla permite hasta 51 caracteres por línea

Otra característica que aparece en la versión 08 y siguientes es el Micro-Print, que permite la impresión con 42 ó 51 columnas en la pantalla de nuestro televisor. Se pueden mezclar los tres formatos de impresión en la misma línea, y el único inconveniente es que los atributos no coinciden exactamente con los límites de los caracteres cuando se trabaja a más de 32 columnas.



La definición de formatos de impresión resulta ser una de las tareas principales en la creación de cualquier Base de Datos, ya que la utilidad del programa depende de la mayor parte de las veces de lo flexible y completa que sea la presentación de esta información, sea en pantalla o impresora. Para editar un formato se debe pulsar E en el menú principal. Tras ello, la opción A permite crear un nuevo formato, y la R repasar o alterar un formato ya existente. Las características generales de cada formato incluyen color de borde y papel, número de líneas entre registros y secuencia de presentación de los registros.

Unos formatos de impresión flexibles son el alma de cualquier Base de datos

Los formatos son bastante generales, ya que permiten mostrar desde 1 hasta 22 registros en cada pantalla, y en cualquier orden, ya que el campo de indexación es arbitrario. A partir de las características generales, se pueden añadir tantos elementos particulares como queramos. Estos pueden ser Literales, es decir, textos en una posición fija de la pantalla, Cajas, que se utilizan para enmarcar datos, y líneas horizontales o verticales, importantes a la hora de tabular nuestro fichero.

Eso en cuanto a los elementos estáticos. Pero lo más importante de cualquier formato son los elementos dinámicos, es decir, aquéllos que referencian datos de nuestro fichero. Para ello hay que decirle al programa qué campo de cada registro debe utilizar, en qué línea de la pantalla se debe comenzar la impresión, si usamos o no Micro-Print, columna de inicio, anchura y número de líneas, atributos de color, si queremos o no que nuestro texto vaya enmarcado con el color de papel elegido y, finalmente, el texto que debe escribir el programa si el campo indicado está ausente de un registro.

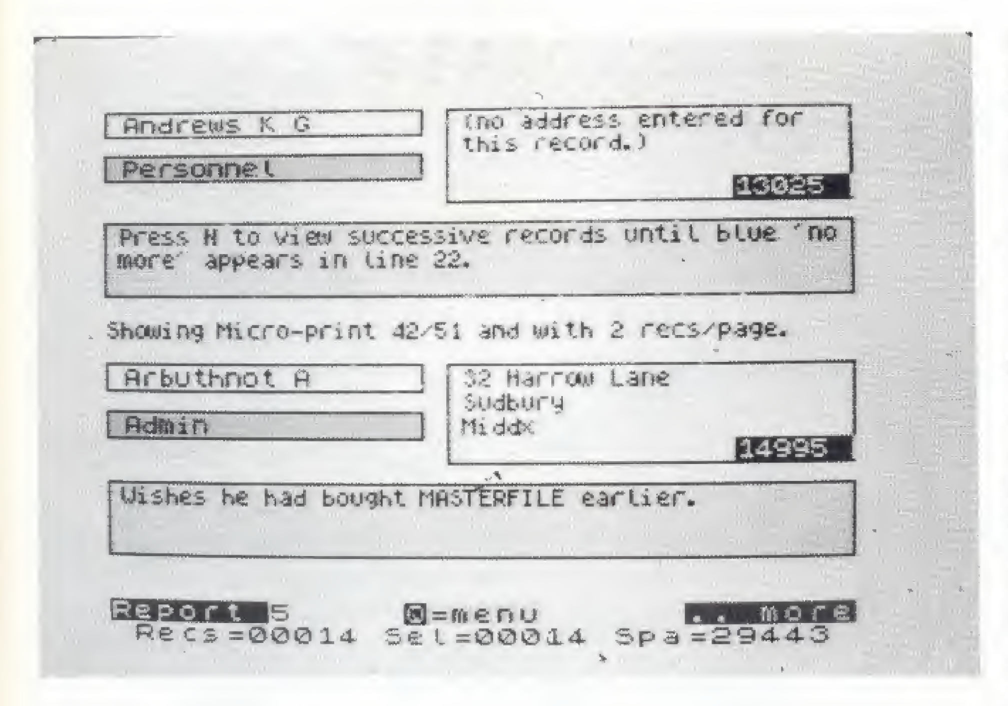
Los formatos resultan muy complicados, pero aprovechan al límite las posibilidades del Spectrum.

Otro elemento delicado de una Base de datos es la facilidad y flexibilidad con que se puede definir la estructura de campos de un fichero. MasterFile resulta potente pero confuso, como en otros casos. Para definir un fichero, nos dice el manual, debemos comenzar por crear un fichero vacío. Esto implica borrar todas las definiciones de formatos y campos de datos.

A continuación, la opción N del menú principal nos permite alterar las definiciones de campos. Como en ese momento no tendremos ninguna activa, debemos pulsar A (añadir) para introducir nombres de campos. Cada campo se identifica por una letra, y se puede acompañar por un nombre de hasta 128 caracteres, lo que nos permitirá recordarlos con más facilidad. Una vez definidos todos los nombres de nuestros datos podemos pasar a definir o alterar, como se mostró anteriormente, los formatos de impresión.

Otras opciones en el menú principal permiten listar todos los formatos eligiendo uno de ellos, cargar y salvar ficheros, invertir la selección de registros y borrar los registros seleccionados. Las restantes opciones son más complicadas, incluyendo las opciones ya comentadas de editar formatos y nombres de campos, además de menús de búsqueda, presentación visual, etc.

El menú de presentación visual, al que se accede pulsando D en el menú principal, permite ver el fichero mediante el primer formato que definimos, o el último que se ha utilizado. La tecla Q muestra las posibles opciones en este modo, bastando otra pulsación de la misma tecla para volver a ver los registros en pantalla. Pulsando N pasamos a la siguiente página de listado.



Las teclas 1-9 permiten avanzar de uno a nueve registros. En la versión 09, la tecla 0 permite volver atrás un registro. B vuelve al comienzo del fichero. Si pulsamos U, el programa pasa a modo edición con el registro que aparece en la parte superior de la pantalla, y E nos permite borrar el registro en la misma posición. La letra O deselecciona el registro en la parte alta de la pantalla, y C crea una copia situándola a continuación.

R se utiliza para cambiar el formato, manteniendo activo el registro superior. Resulta muy útil para buscar mediante un formato con 20 registros por página y, una vez hemos situado en posición el registro deseado, cambiar a un formato más amplio, que muestre la totalidad de los datos. En este momento nos resulta útil saber que, pulsando P, se obtiene una copia de la pantalla, y que, si a continuación tecleamos A, cada nueva página que se muestre irá a la impresora.

Búsqueda en nuestro fichero

De poco sirve tener un fichero en el ordenador si tenemos que repasar registro a registro para encontrar el dato que nos interesa. Por ello, el menú principal dispone de la opción S para buscar y seleccionar parte de las fichas de nuestro archivo. Primero debemos indicarle si queremos que seleccione a partir de todo el fichero o bien a partir de las fichas que habíamos seleccionados previamente. Después el programa nos pregunta el campo en el que debe realizar la selección, y si ésta debe ser numérica o alfanumérica. Para cada caso existen varios tipos de comparaciones: mayor, igual, menor, distinto y, en el caso de caracteres, ocurrencia de una palabra en un campo. Para finalizar la búsqueda, el programa nos pregunta qué cadena o número debe utilizar para la comparación, y nos devuelve el número de fichas seleccionadas tras la búsqueda.

Se pueden seleccionar y ordenar ficheros con todas las opciones razonables.

Otra opción interesante del programa es la opción T, que calcula las sumas y promedios de los campos numéricos del fichero. Si durante el proceso se encuentra algún campo no numérico, el programa nos da la opción de ignorarlo o corregir el valor. Si el campo está ausente, se le asigna un valor de cero para calcular el promedio. Sólo se pueden acumular los valores de un campo mediante este comando, por lo que, si queremos realizar operaciones más complicadas deberemos introducir un programa Basic que haga la operación requerida.

Incluyendo un programa Basic, MasterFile se convierte en un programa a la medida

A pesar de la versatilidad del programa, existen casos en que resulta muy útil realizar cálculos con datos de varios campos de un mismo registro, o insertar campos con un valor determinado en función del valor de otros. MasterFile permite ambas posibilidades, y lo hace mediante un esquema sencillo y eficaz. Pulsando U en el menú principal se obtiene el acceso al programa del usuario, que debe seguir el siguiente esquema:

En primer lugar, el programa le da control a la línea 4900, una sola vez, antes de procesar el fichero. Este es un lugar interesante para aquellas tareas, como inicialización de variables, que sólo deben hacerse una vez. Como el resto de los bloques, debe acabarse esta parte con la instrucción GOTO USR R.

A continuación se le da control a la línea 5000 una vez para cada registro, antes de procesar los campos; esto permite inicializar las variables que varíen en cada registro.

La línea 6000 se ejecuta un vez por cada campo. Al entrar, la variable C$ contiene el campo, con la referencia en C$(1) y los datos en C$(2 to).

La línea 7000 se ejecuta cuando se han procesado todos los campos del registro, y en ella se puede crear un nuevo campo o no, según el valor que le demos a C$. Si C$="", el registro queda como estaba; si C$(1) es un referencia de campo, y c$(2 to) está vacío, entonces el programa borra el campo indicado; si C$(1) es una referencia de campo y c$(2 to) contiene datos, el campo indicado se crea o actualiza. Como en los demás casos, el retorno al programa se realiza mediante GOTO USR R.

La impresión se realiza siempre a través de la instrucción COPY.

En la versión 9 y siguientes la línea 9000 se ejecuta después de procesar todo el fichero, lo que permite imprimir los resultados finales o poner a cero algunas variables. El orden de paso de los registros a través del programa de usuario es el de inserción.

Adaptación a microdrive o impresora

Para facilitar la adaptación a *microdrive* o *diskette* del programa, todas las operaciones de entrada/salida se realizan desde BASIC. Por tanto, el programa resulta fácilmente adaptable, ya que sus operaciones de entrada/salida se reducen a variables, código máquina o el propio programa. Por ello, el programa no saca provecho de las posibilidades de acceso aleatorio que ofrecen estos dispositivos.

La impresión se realiza siempre a través de la instrucción COPY. El programa funcionará correctamente sólo con *Interfaces* que realicen copias de pantalla. Además, el código máquina de impresora no debe residir en la misma zona de memoria que el programa, por lo que la compatibilidad resulta más difícil. Un consejo: prueba antes de comprar el programa que tu impresora podrá obtener resultados con él.

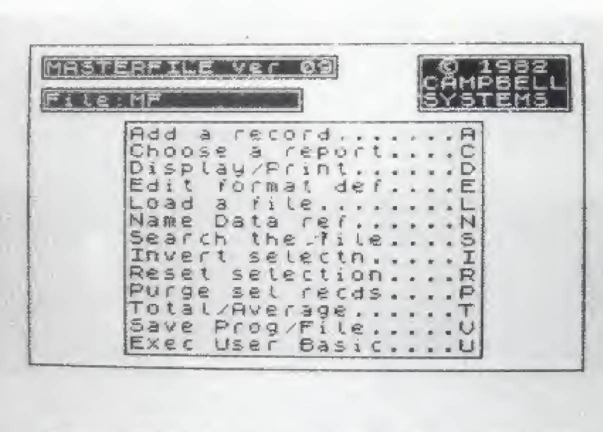
GUSANEZ

por José C. Tomás

```
Name and address labels, 3-UP
Mr.Humphreys
Lives with his Mother.

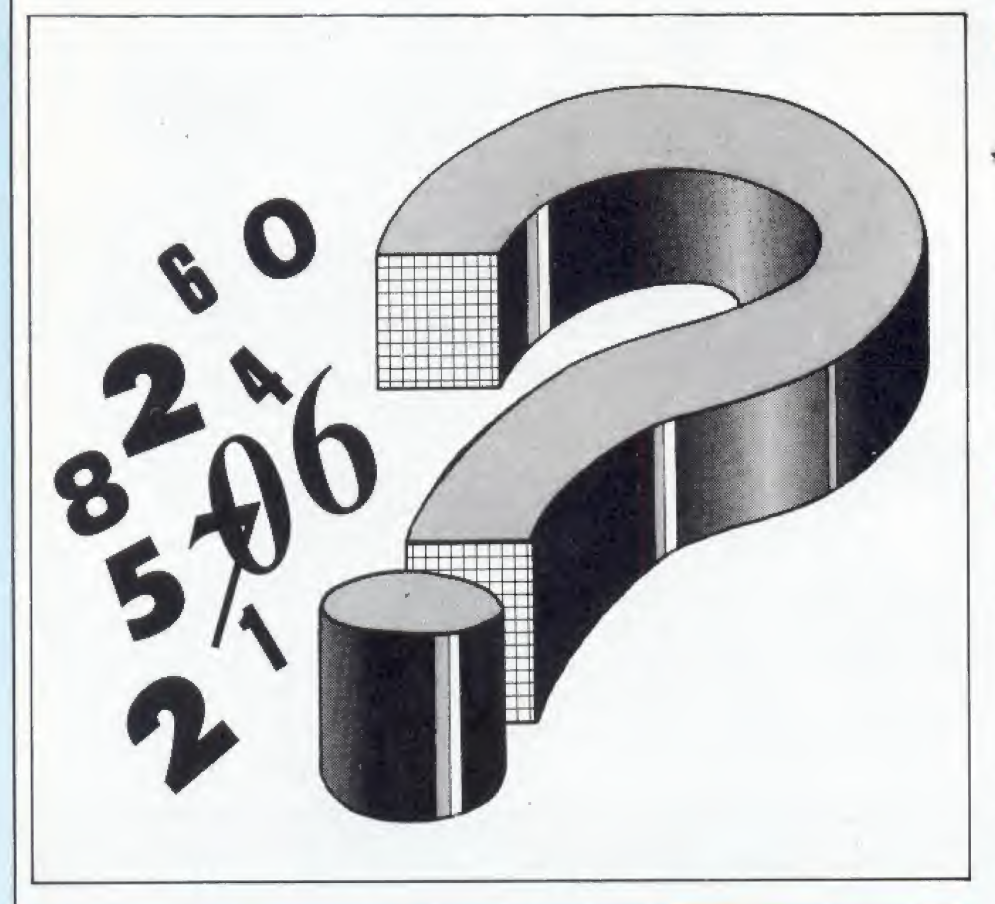
Printer
No address has been
provided, so we have
used a Null Text
filler instead.

Programmer A
Duncoding
Memory Lane
ROMford
Z80 4MHZ
Report 4        =menu        more
Recs=00014 Sel=00014 Spa=29443
```



Una buena Base de datos, algo difícil de manejar

Sin ánimo de exagerar, se puede decir que MasterFile es una de las mejores Bases de datos que existen para Spectrum. Por otra parte, su uso resulta complicado por la variedad de opciones y la estructura de menús, que hace difícil saber en qué opción del programa nos encontramos. Para quienes no tengan miedo de estudiar cuidadosamente las instrucciones y experimentar, MasteFile les ofrece la posibilidad de gestionar sus ficheros de datos de una manera muy flexible.



Factoriales con todas las cifras

En teoría de probabilidades y en muchas otras aplicaciones es muy frecuente la utilización de factoriales. Este sencillo cálculo, que se define como n! = n * (n–1) * (n–2) *... * 2 * 1, tiene la característica de desbordar rápidamente la capacidad del ordenador. En efecto, el factorial de un número tan pequeño como 30 da la desorbitante cifra de 2,65 * 10 ↑ 32. El de 34 provoca ya un error del tipo «Number too big» si intentamos calcularlo sin recurrir a ninguna argucia. Esto no sólo ocurre en el Spectrum sino en cualquier otro ordenador. Por si a alquien le consuela, el famoso PC de IBM (con un procesador 8088 de 16 bits) también falla al llegar a 34, la única diferencia es que en este caso el informe es «Overflow». El Commodore 64 se queda igualmente en 34, mientras que los MSX van algo más lejos: la palabra «Overflow» aparecer al llegar a 49. El programa utilizado para esta sencilla prueba es:

```
10 LET C=1
15 FOR N=1 TO 100
20 PRINT N
25 LET C=C*N
30 NEXT N
```

Con el QL de Sinclair tuvimos que «alargar» el bucle FOR-NEXT hasta 1000, pues batió todas las marcas llegando sin problemas hasta el factorial de 301.

Del mismo modo que un ordenador sabe multiplicar porque tiene en su memoria ROM un programa en código máquina que le indica cómo hacerlo, también podemos enseñarle a calcular factoriales de números considerablemente elevados. Sólo se necesita el programa adecuado. Con el que presentamos en estas páginas, el máximo alcanzable no depende tanto del Spectrum como de nuestra propia paciencia. Por ejemplo, invierte 3 minutos 15 segundos en calcular las 158 cifras de 100! y unos 70 minutos para las 615 de 300!. El método empleado se basa en la aproximación de Stirling, matemático inglés del siglo XVIII famoso por su fórmula para el cálculo de factoriales:

$$n! \approx n^n e^{-n} \sqrt{2\pi n}$$

El programa obtiene primero el número de ceros de n! mediante la subrutina que comienza en la línea 340. A continuación, calcula el número de cifras utilizando el logaritmo de la aproximación de Stirling y, por último, halla el resultado mediante una serie de bucles FOR-NEXT que almacenan los resultados parciales en la matriz b. El resultado final puede almacenarse en la variable alfanumérica n$.

```
  1 REM         FACTORIAL
  2 REM
  3 REM   CON TODAS LAS CIFRAS
  4 REM
  5 REM       SPECTRUM 16/48K
  6 REM
  7 REM       SEPTIEMBRE 1984
  8 REM
  9 REM
 10 REM   CONSTANTINO CORTINA
 11 REM
 12 REM
 20 INPUT "Factorial de ? ";n
 30 IF n<0 OR n-INT n<>0 THEN
GO TO 20
 40 LET nf=n: GO SUB 350
 50 LET g=INT (1+nc/6)
 60 REM c= Logaritmo de la
        aproximacion de Stirling
 70 LET c=INT (((n+.5)*LN n-n+.
5*LN (2*PI))/LN 10)
 80 LET k=INT (1+c/6)
 90 DIM a(k): DIM b(k)
100 LET a(1)=1: LET p=1
110 FOR i=1 TO k
120 LET a(i)=a(i)*p: IF a(i)=0
AND i>g THEN  GO TO 150
130 LET b(i)=INT (a(i)/10e5)
140 NEXT i
150 FOR i=2 TO k
160 LET a(i)=a(i)+b(i-1)
170 NEXT i
180 FOR i=1 TO k
190 LET a(i)=a(i)-10e5*INT (a(i
)/10e5)
200 NEXT i
210 IF p<>n THEN  LET p=p+1: GO
 TO 110
220 BEEP .1,0
230 PRINT n;"!="
240 FOR i=k TO 1 STEP -1
250 LET a$=STR$ a(i)
260 IF i=k THEN  GO TO 300
270 LET a$="00000"+a$
280 LET j=LEN a$
290 LET a$=a$(j-5 TO j)
300 PRINT a$
310 NEXT i
320 INPUT "Quieres guardarlo en
n$ ? ";b$
330 IF b$(1)="s" THEN  GO SUB 5
00
335 PRINT : GO TO 20
340 REM nc= Numero de ceros
            de n!
350 LET nc=0
360 LET co=INT (nf/5)
370 IF co=0 THEN  RETURN
380 LET nc=nc+co
390 LET nf=co
400 GO TO 360
500 LET n$=STR$ n+"!="
510 FOR i=k TO 1 STEP -1
520 LET a$=STR$ a(i)
530 IF i=k THEN  GO TO 570
540 LET a$="00000"+a$
550 LET j=LEN a$
560 LET a$=a$(j-5 TO j)
570 LET n$=n$+a$
580 NEXT i
590 PRINT 'n$
600 RETURN
```

```
100!=93326215443944152681699238
856266700490715968264381621468592
963895217599993229915608941463976
156518286253697920827223758251
185210916864000000000000000000000
000
```



Nuestro amigo Gusánez está ya harto de merodear por la revista y ha decidido largarse a la playa con su amigo el «enano». Pero para ello necesita vuestra ayuda: tenéis que guiar sus pasos para que encuentre a su amigo y sortear los diversos peligros que encontrará en su camino. Sólo vosotros podéis conseguir que Gusánez se vaya de vacaciones y nos deje respirar un poco en la redacción.

Gusánez se va de vacaciones

Concretamos un poco más: para que Gus pueda salir airoso de su aventura debe recoger seis objetos:

Microordenador Spectrum.
Maleta con la inicial G.
Unas gafas de sol.
Unas gafas de bucear.
Un radio-cassette.
Una revista TODOSPECTRUM.

Además nuestro querido amigo tiene que encontrar a su querido amigo. Las formas de encontrarlo son variadas. Nos explicamos.

Forma 1: El enano se mueve constantemente por las distintas pantallas, pero el Gus no lo sabe. Si logra encontrar su pista y seguirla podrá hacerse con él.

Forma 2: Logra que el enano, cuando se presente de improviso, logre cruzar de arriba abajo la pantalla a través de unos bloques que el Gus podrá mover a izquierda y derecha.

La única forma de conseguir la posición del enano en el laberinto es consultar el mapa.

Pero igual que varias formas de encontrar al enano, hay una de

51

Los caracteres que aparecen subrayados en el listado corresponden a caracteres gráficos del ordenador.

```
  10 BORDER 5: PAPER 7: INK 2: B
RIGHT 0: CLS : RANDOMIZE 0
  20 POKE 23658,0: GO SUB 3730
  30 LET cr=0: LET vb=0: LET nm=
0: LET x1=25: LET x2=1: LET x3=1
8: LET x4=3: LET di=0
  40 LET w$="": LET nx=0: LET ch
=0: LET d=300: LET d1=0: DIM a(2
0,20): LET a=18: LET b=23
  50 LET o$="M": DIM p$(7): LET
ro=1: LET co=2: LET a1=0: LET b1
=0
  60 LET px=0: LET ti=15: LET t=
0: LET pa=7: LET in=1: LET ob1=0
: LET en=0: LET w=63: LET g$="AB
": LET h$=" CD"
  70 GO SUB 3780: GO SUB 1530
  80 GO SUB 3710
  90 LET w=63: GO SUB 2690: GO S
UB 750
 100 GO SUB 480
 110 LET e$=INKEY$
 120 IF INKEY$<>"" THEN  LET e$=
INKEY$
 130 IF e$="z" THEN  GO SUB 320:
 GO TO 120
 140 IF e$="x" THEN  GO SUB 360:
```



****** LAS VACACIONES DE *******

* GUSANEZ SE MUEVE CON Z,X,P y L
SE PARA CON OTRA TECLA

* COGE LOS 6 OBJETOS DIFERENTES

* BUSCA AL ENANO Y CUIDADO CON
LOS GRANDES ALMACENES

* CONVIERTE A GUS EN UN COCHE
COGIENDO LA ULTIMA FLOR Y...
ACERTANDO LA CLAVE SECRETA

* BEBE REFRESCOS POR EL CAMINO

* TU META ES LLEGAR A LA PLAYA

Pulsa una tecla

perderlo: Los grandes almacenes. Aquí si no logra el Gus encontrar el objeto escondido se quedará sin su amigo.

Una vez tengas los seis objetos y el enano, y superados los puntos propuestos, deberás acertar la clave secreta. Esta, a través del juego, se ha ido visualizando en la parte superior de la pantalla letra a letra

Programas

51

pero sin ningún orden e incluso repetidas. Para acertar la palabra correcta deberás, en un tiempo determinado, pulsar letra a letra hasta completarla. Si te equivocas pulsa DELETE y empieza de nuevo aunque el tiempo sigue su curso.

Gusánez comienza a perder puntos en cuanto se le sitúa sobre una superficie oscura.

Si acierta la clave el Gus se convierte en coche y empezará a gastar los puntos acumulados hasta el momento. En el indicador de posición se verá un punto parpadeante

```
 GO TO 120
 150 IF e$="p" THEN  GO SUB 400:
 GO TO 120
 160 IF e$="l" THEN  GO SUB 440:
 GO TO 120
 170 IF (p<>p1 OR o<>o1) AND ch=
0 AND a(p,o)<=0 THEN  LET ti=ti-
1: PRINT  BRIGHT 1;AT 19,29;ti;"
 ": BEEP .01,p+40: IF ti<=0 THEN
  GO SUB 1410
 180 IF g$=" GF" AND (ATTR (a,b+
1)<>58 OR ATTR (a,b+2)>58 OR ATT
R (a-1,b)>58 OR ATTR (a-1,b+1)>5
8) THEN  GO SUB 500
 190 IF g$="AB " AND (ATTR (a,b)
<>58 OR ATTR (a,b+1)<>58 OR ATTR
 (a-1,b+1)<>58 OR ATTR (a-1,b+2)
<>58) THEN  GO SUB 500
 200 LET en1=RND*100: IF en1>90
THEN  GO SUB 3120
 210 IF g$="AB " AND a=a1 AND b+
2=b1 THEN  GO SUB 2210
 220 IF g$=" GF" AND a=a1 AND b=
b1 THEN  GO SUB 2210
 230 GO TO 110
 240 IF ch=1 THEN  RETURN
 250 LET nu=INT (3+RND*7)
 260 FOR n=1 TO 19: PRINT AT n,0
  INK 0;"[redacted]": NEXT n
 270 PRINT  FLASH 1;AT 21,13;"CO
GE LA ULTIMA": FOR n=1 TO nu: BE
EP .01,45+n: LET c1=INT (4+RND*1
4): LET d1=INT (2+RND*26): FOR m
=7 TO 2+RND*4 STEP -1: PRINT  IN
VERSE 1; PAPER 1; INK m;AT c1,d1
;"J": NEXT m: NEXT n
 280 LET a1=c1: LET b1=d1: GO SU
B 650
 290 IF d>=to THEN  GO SUB 570
 300 RETURN
 310 REM  Movimiento gusanez
 320 IF ATTR (a,b)=120 THEN  GO
SUB 530
 330 IF ATTR (a,b)>=66 OR ATTR (
a-1,b-1)>=66 THEN  GO TO 170
 340 IF b<=0 THEN  BEEP .01,20:
LET p=p-1: LET b=x1: GO SUB 750:
 GO SUB 480
 350 GO SUB 480: LET g$=" GF": L
ET h$=" IH ": LET b=b-1: GO SUB 4
80: RETURN
 360 IF ATTR (a,b+2)=120 THEN  G
O SUB 550
 370 IF b>=26 THEN  BEEP .01,20:
 LET p=p+1: LET b=x2: GO SUB 750
: GO SUB 480
 380 IF ATTR (a,b+2)>=66 OR ATTR
 (a-1,b+3)>=66 THEN  GO TO 170
 390 GO SUB 480: LET g$="AB ": L
ET h$=" CD": LET b=b+1: GO SUB 4
80: RETURN
 400 IF h$(1)="I" THEN  IF ATTR
(a-2,b)>=66 OR ATTR (a-2,b+1)>=6
6 OR ATTR (a-1,b+2)>=66 THEN  GO
 TO 170
 410 IF h$(1)=" " THEN  IF ATTR
(a-1,b)>=66 OR ATTR (a-2,b+1)>=6
6 OR ATTR (a-2,b+2)>=66 THEN  GO
 TO 170
 420 IF a<=2 THEN  BEEP .01,20:
LET o=o-1: LET a=x3: GO SUB 750:
 GO SUB 480
 430 GO SUB 480: LET a=a-1: GO S
UB 480: RETURN
 440 IF h$(1)=" " THEN  IF ATTR
(a+1,b)>=66 OR ATTR (a+1,b+1)>=6
6 OR ATTR (a,b+2)>=66 THEN  GO T
O 170
 450 IF h$(1)="I" THEN  IF ATTR
(a,b)>=66 OR ATTR (a+1,b+1)>=66
OR ATTR (a+1,b+2)>=66 THEN  GO T
```

al cual te deberás dirigir.

Si no aciertas dicha clave el ordenador elegirá otra nueva y modificará los puntos a los que tendrás que llegar de nuevo.

Con estas instrucciones podrás llegar a la playa, aunque hay muchas más cosas que tendrás que descubrir junto a tu amigo Gusánez. Felices vacaciones.

José Carlos Tomás **48K**

Para coger objetos basta situarlos bajo la narizota de Gusánez y detener su movimiento.

```
0 180
 460 IF a>=19 THEN  BEEP .01,20:
 LET o=o+1: LET a=x4: GO SUB 750
: GO SUB 480
 470 GO SUB 480: LET a=a+1: GO S
UB 480: RETURN
 480 IF ch=1 THEN  LET d1=50: GO
 SUB 1950
 490 PRINT  PAPER 8; FLASH 8; BR
IGHT 8; INK 8; OVER 1;AT a,b;g$;
AT a-1,b;h$: BEEP .0001,60: RETU
RN
 500 LET d=d-50: PRINT AT 21,7;d
;" ": BEEP .01,a: BEEP .01,b: FO
R n=0 TO 5: BORDER n: NEXT n
510 IF d<=0 THEN FOR n=1 TO 60:
 BEEP .01,n: PRINT OVER 1;AT 21,
0;"[redacted]": NEXT n: GO TO
3550
 520 RETURN
 530 IF b-4<=1 THEN  RETURN
 540 FOR n=b-5 TO b-6 STEP -1: P
RINT AT a,n;"RRRRR ": BEEP .001,
n+30: NEXT n: PRINT  BRIGHT 1; I
NK 0; PAPER 7;AT a,n+1;"RRRRR";
BRIGHT 0;AT a,n+3;" ": RETURN
 550 IF b+7>=29 THEN  RETURN
 560 FOR n=b+2 TO b+4: PRINT AT
a,n;" RRRRR": BEEP .001,n+30: NE
XT n: PRINT  BRIGHT 1; INK 0; PA
PER 7;AT a,n;"RRRRR"; BRIGHT 0;A
T a,n+2;" ": RETURN
 570 REM  Puertas
 580 IF ch=1 THEN  RETURN
 590 LET pu1=INT (2+RND*16): FOR
 n=pu1 TO pu1+1: PRINT AT n,0;"
": NEXT n: BEEP .01,pu1
 600 LET pu2=INT (2+RND*16): FOR
 n=pu2 TO pu2+1: PRINT AT n,28;"
 ": NEXT n: BEEP .01,pu2
 610 LET pu3=INT (2+RND*16): PRI
NT AT 1,pu3;"    ": BEEP .01,pu3
 620 LET pu4=INT (2+RND*16): PRI
NT AT 19,pu4;"    ": BEEP .01,pu4
```

GUSANEZ

por José C. Tomás

Programas


```
 630 GO SUB 990
 640 GO SUB 1200: RETURN
 650 IF ch=1 THEN  RETURN
 660 LET pa=INT (RND*7): FOR n=1
 TO 19: PRINT AT n,0; BRIGHT 1;
PAPER pa;"F";AT n,28;"A": NEXT n
 670>PRINT AT 1,1; BRIGHT 1; PAP
ER pa;"█████████████████████████
███";AT 19,1;"██████████████████
███████"
 680 RETURN
 690 IF ch=1 THEN  RETURN
 700 GO SUB 900: LET di=0: FOR n
=1 TO 19: PRINT AT n,0; INK 6; P
APER 5; BRIGHT 1;"R";AT n,28;"R"
: NEXT n
 710>PRINT AT 1,1; INK 6; PAPER
5; BRIGHT 1;"RRRRRRRRRRRRRRRRRR
RRRRRRRR";AT 19,1;"RRRRRRRRRRRRR
RRRRRRRRRRRRR"
 720 PRINT  FLASH 1;AT 21,13;"CO
GE LA ULTIMA": FOR m=1 TO p: LET
 a1=INT (3+RND*15): LET b1=INT (
2+RND*25): FOR n=5 TO 7: BEEP .0
1,n-10: PRINT  OVER 1; INK n;AT
a1,b1;"O": NEXT n: NEXT m: LET o
$="J"
 730 RETURN
 740 REM  Pantalla laberinto
 750 LET px=0: LET cr=0: LET di=
0: GO SUB 3120: GO SUB 900
 760 IF a(p,o)=20 THEN  GO TO 24
20
 770 IF p=p1 AND o=o1 THEN  LET
b=24: GO SUB 2740: GO SUB 990: R
ETURN
 780 GO SUB 1040
 790 IF a(p,o)=40 THEN  GO SUB 3
430: RETURN
 800 IF a(p,o)=1 THEN  GO SUB 32
50: GO SUB 1070: GO SUB 990: RET
URN
 810 PRINT  INVERSE 1;;AT 20,0;"
----> ";to;" puntos ";"<-----"
 820 IF a(p,o)=8 THEN  GO SUB 16
20: GO SUB 650: GO SUB 570
 830 IF (a(p,o)=-1 OR a(p,o)=-2)
 AND ch=0 THEN LET in=RND*7: LET
 pa=RND*7: FOR n=1 TO 19: PRINT
AT n,0; INK in; PAPER pa;"RRRRRR
RRRRRRRRRRRRRRRRRRRR": NEXT n
: GO SUB 660: GO SUB 570: GO SUB
 1320: RETURN
 840 IF a(p,o)=-5 THEN  GO SUB 6
50: GO SUB 570: RETURN
 850 IF a(p,o)>=9 THEN  LET q$=C
HR$ (65+nm): GO SUB 240
 860 IF a(p,o)=-3 THEN  GO SUB 6
90: GO SUB 570: RETURN
 870 IF a(p,o)=0 THEN  GO SUB 65
0: GO SUB 1340: GO SUB 570: RETU
RN
 880 IF a(p,o)<8 AND a(p,o) THEN
 GO SUB 650: GO SUB 570: GO SUB
 1070
 890 RETURN
 900 FOR n=1 TO 21: PRINT AT n,0
;"                              "
: NEXT n
 910 LET ti=15: FOR n=17 TO 21:
PRINT AT n,29; BRIGHT 1; INK n-1
6;" ███": NEXT n
 920 IF p=a3 AND o=b3 THEN  LET
en=1: LET t=1
 930 PRINT AT 21,0; INK 6; PAPER
 1; BRIGHT 1;"PUNTOS:";d: FOR n=
14 TO 16: PRINT AT n,29;"   ": N
EXT n
 940 PLOT 232,40: DRAW 23,0: DRA
W 0,23: DRAW -23,0: DRAW 0,-23
 950 PLOT 233+p,62-o: PLOT  INK
0;233+p1,62-o1
 960 IF en=1 THEN  PRINT AT 17,3
0; FLASH 1; BRIGHT 1;"F"
 970 IF ch=1 THEN  PLOT  FLASH 1
;233+a2,62-b2: GO SUB 990
 980 RETURN
 990>IF o<=1 THEN LET cr=cr+1: P
RINT BRIGHT 1;AT 1,0;"0000000000
0000000000000000000000"
```





¡APROVECHA AL MAXIMO TU SPECTRUM!

Ahora, a tu alcance, dos obras fundamentales para que podáis sacar todo el partido posible a vuestro ordenador.



Esta publicación está diseñada para guiar al nuevo usuario del ZX Spectrum desde el momento que el ordenador se conecta hasta conseguir una base suficiente de la programación BASIC.
Incluye temas como:

- Introducción al teclado.
- Instrumentos útiles para la programación.
- Uso de comandos fáciles.
- Como construir un programa.
- Técnicas de programación.
- Aplicaciones prácticas.

100 pags. - 750 PTAS.



Este libro, escrito en estilo ameno y práctico, está dirigido a todos aquellos usuarios que han dejado atrás la etapa de los juegos y necesitan adentrarse en el fabuloso mundo de la programación.
El temario incluye:

- Reglas y herramientas del BASIC.
- La técnica de los organigramas.
- Cómo planificar un programa.
- El mundo de las rutinas.
- Variables y cadenas.
- Funciones matemáticas usuales.

109 pags. - 750 PTAS.

CUPON DE PEDIDO

Recorta este cupón debidamente cumplimentado y envíelo a INFODIS, S. A. C/ BRAVO MURILLO, 377-5.° A - 28020 MADRID

Si, envíenme el(los) libro(s) que a continuación detallo al precio de 750 ptas. libro, más 100 ptas. en concepto de gastos de embalaje y envío.

El importe lo abonaré: POR CHEQUE ☐ CONTRAREEMBOLSO ☐ CON TARJETA DE CREDITO (VISA ☐) (AMERICAN EXPRESS ☐) (INTERBANK ☐)

Número de mi tarjeta

TITULO ________________

____

NOMBRE ________________

CALLE ________________

CIUDAD ________________ D. P. ________________

PROVINCIA ________________

Firma

```
1000 IF p<=1 THEN  LET cr=cr+1:
FOR n=1 TO 19: PRINT  BRIGHT 1;A
T n,0;"@": NEXT n
1010 IF o>=20 THEN  LET cr=cr+1:
PRINT BRIGHT 1;AT 20,0;"@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@"
1020 IF p>=20 THEN  LET cr=cr+1:
 FOR n=1 TO 19: PRINT  BRIGHT 1;
AT n,28;"@": NEXT n
1030 RETURN
1040 FOR n=w TO w-2 STEP -1: PLO
T  OVER 1; INK 8; PAPER 8;232,n:
 DRAW  OVER 1; INK 8; PAPER 8;2,
0: BEEP .01,60: NEXT n: LET w=w-
3: IF w<64 THEN  GO SUB 2900
1050 IF w=80 THEN  FOR n=1 TO 20
: BEEP .05,30: PRINT  OVER 1;AT
0,12;"T T T T T T T": NEXT n
1060 RETURN
1070 REM  Objeto
1080 IF a(p,o)=1 THEN  LET o$="T
": LET a1=16: LET b1=INT (11+RND
*8): PRINT  PAPER 4; INK 2;AT a1
,b1;"T": RETURN
1090 IF ch=1 THEN  RETURN
1100 IF a(p,o)=0 THEN  GO SUB 13
40: RETURN
1110 GO SUB 3380
1120 IF a(p,o)=2 THEN  LET o$="O
": PRINT  INK 1;AT a1,b1;o$
1130 IF a(p,o)=3 THEN  LET o$="N
": PRINT  INK 3;AT a1,b1;o$
1140 IF a(p,o)=4 THEN  LET o$="M
": PRINT AT a1,b1;o$
1150 IF a(p,o)=5 THEN  LET o$="L
": PRINT  INK 4;AT a1,b1;o$
1160 IF a(p,o)=6 THEN  LET o$="K
": PRINT  INK 5;AT a1,b1;o$
1170 IF a(p,o)=7 THEN  LET o$="U
": PRINT  INK 1;AT a1,b1;o$
1180 IF a3=p AND b3=o THEN  LET
en=1: PRINT AT a-1,b;" "
1190 BEEP .05,30: RETURN
1200 IF en=1 THEN  LET br=0
1210 IF en=0 THEN  LET br=1
1220 IF p=1 THEN  GO TO 1240
1230 IF a(p-1,o)=-5 THEN  LET cr
=cr+1: FOR n=pu1 TO pu1+1: PRINT
  BRIGHT br; INK 1;AT n,0;"P": N
EXT n: BEEP .01,pu1
1240 IF p=20 THEN  GO TO 1260
1250 IF a(p+1,o)=-5 THEN  LET cr
=cr+1: FOR n=pu2 TO pu2+1: PRINT
  BRIGHT br; INK 1;AT n,28;"P":
NEXT n: BEEP .01,pu2
1260 IF o=1 THEN  GO TO 1290
1270 IF a(p,o-1)=-5 THEN  LET cr
=cr+1: PRINT  BRIGHT br; INK 1;A
T 1,pu3;"PPP": BEEP .01,pu3
1280 IF o=20 THEN  RETURN
1290 IF a(p,o+1)=-5 THEN  LET cr
=cr+1: PRINT  BRIGHT br; INK 1;A
T 19,pu4;"PPP": BEEP .01,pu4
1300 IF cr>=4 AND br=1 AND ch=0
THEN  GO SUB 480: PRINT AT 11,10
; FLASH 1;"!ATRAPADO!": BEEP 5,-
40: GO TO 3550
1310 LET cr=0: RETURN
1320 PLOT 8,88: DRAW  BRIGHT 1;
INK in; PAPER in;208,0,-2.4+(RND
*4.8)
1330 RETURN
1340 LET ds=INT (RND*3): IF NOT
ds THEN  GO TO 1380
1350 LET dx=INT (8+RND*210): PLO
T dx,24: DRAW  INK 7; BRIGHT 1;0
,132
1360 BEEP .05,50: LET dy=INT (24
+RND*128): PLOT 8,dy: DRAW  INK
7; BRIGHT 1;210,0
1370 BEEP .05,55: RETURN
1380 LET px=1: LET dz=INT (2+RND
*24): PRINT  BRIGHT 1; INK 1; PA
PER 7;AT 2,dz;"P";AT 3,dz-3;"A
 F";AT 19,dz-3;"A   F": FOR n=4
TO 18: LET dy=INT (4+RND*17)
1390 PRINT  BRIGHT 1; INK 0; PAP
ER 7;AT n,dy;"RRRRR"; BRIGHT 0;A
T n,dy+2;" ": BEEP .003,38: NEXT
 n
1400 RETURN
1410 IF px=1 THEN  LET px=0: GO
TO 1440
1420 IF a(p,o)=-3 THEN  LET d1=1
000: GO SUB 1950: LET ti=15: LET
 a(p,o)=INT (RND*20): RETURN
1430 IF a(p,o)<>-3 THEN  LET ti=
15: GO SUB 3050: RETURN
1440 PRINT AT 2,dz; OVER 1;"P":
FOR n=3 TO 17
1450 PRINT AT n,dz; BRIGHT 1;"P"
: BEEP .1,n: PRINT  BRIGHT 1; OV
ER 1;AT n,dz;"P"
1460 IF ATTR (n+1,dz)<>120 THEN
```

```
 GO SUB 1510: RETURN
1470 NEXT n
1480 FOR n=1 TO 60 STEP 3: PRINT
  OVER 1;AT 19,dz;"P": BEEP .01,
22: NEXT n
1490 IF en=1 THEN  LET d1=5000:
GO SUB 1920: LET ti=15: RETURN
1500 LET ti=15: LET en=1: LET t=
1: GO SUB 920: RETURN
1510 FOR m=dz TO 27: PRINT  OVER
 1;AT n+1,m;"P": BEEP .001,m: PR
INT  OVER 1;AT n+1,m;"P": NEXT m
1520 LET d1=(n-2)*100: GO SUB 19
20: LET ti=15: LET a(p,o)=-1: RE
TURN
1530 REM  Matriz laberinto
1540 FOR m=1 TO 20: FOR n=1 TO 2
0
1550 LET a(m,n)=INT (-3+RND*23)
1560 BORDER 1: BORDER 2: BORDER
4: BORDER 6
1570 NEXT n: NEXT m
1580 FOR n=1 TO 10+RND*30: LET a
(INT (1+RND*20),INT (1+RND*20))=
40: BEEP .01,25: BORDER 1: BORDE
R 2: BORDER 3: BORDER 4: BORDER
0: NEXT n: BORDER 5
1590 LET a3=INT (1+RND*20): LET
b3=INT (1+RND*20): LET a(a3,b3)=
22: IF a(a3,b3)<22 THEN  GO TO 1
590
1600 LET p=INT (1+RND*20): LET o
=INT (1+RND*20): LET p1=p: LET o
1=o: LET a(p1,o1)=30: IF a(p,o)<
30 THEN  GO TO 1600
1610 PRINT #1; FLASH 1;"Pulsa un
a tecla": PAUSE 0: CLS : RETURN
1620 REM  Pantalla laberinto
1IF ch=1 THEN  RETURN
1640 LET bb=RND*7: LET cc=RND*7:
 PRINT  INK 4;AT 11,1;" QUIERES
VER EL MAPA? S/N "; FLASH 1;AT 1
3,11;"500 Puntos"
1650 IF INKEY$="s" THEN  BORDER
5: BEEP .1,50: LET d1=500: GO SU
B 1950: GO TO 1680
1660 IF INKEY$="n" THEN  BORDER
5: BEEP .2,-40: GO SUB 900: RETU
RN
1670 BORDER bb: BORDER cc: PAUSE
 1: GO TO 1650
1680 GO SUB 900
1690 GO SUB 1870
1700 FOR m=px TO py: FOR n=ox TO
 oy
1710 PRINT AT n,m+3; INK 6; BRIG
HT 1;" "
1720 IF a(m,n)=1 THEN  PRINT  IN
K 3; BRIGHT 1;AT n,m+3;"T"
1730 IF a(m,n)=-3 THEN  PRINT  B
RIGHT 1;AT n,m+3;"*"
1740 IF a(m,n)=-5 THEN  PRINT  I
NK 1; FLASH 1; BRIGHT 1;AT n,m+3
;"R"
1750 IF a(m,n)>1 AND a(m,n)<8 TH
EN  PRINT  BRIGHT 1;AT n,m+3;"o"
1760 IF a(m,n)=8 THEN  PRINT  BR
IGHT 1; INK 4;AT n,m+3;"X"
1770 IF a(m,n)=40 THEN  PRINT AT
 n,m+3;"S"
1780 NEXT n: NEXT m
1790 PRINT  BRIGHT 1; FLASH 1;AT
 o,p+3;"G"
1800 IF en=0 THEN  PRINT  BRIGHT
 1; FLASH 1;AT b3,a3+3;"P"
1810 FOR n=1 TO 5: BEEP .1,33: N
EXT n
1820 LET t$="                /****
Muchos puntos/      /oooo Objeto/
      /RRRR Pista del enano/      /
XXXX Ver mapa/      /TTTT Refresc
o/     /SSSS Grandes almacenes/
      //PULSA UNA TECLA//        "
1830 FOR n=1 TO LEN t$-16
1840 IF INKEY$<>"" THEN  GO TO 1
860
1850 PAUSE 5: PRINT  INK 9; PAPE
R RND*7; BRIGHT 1;AT 21,12;t$(n
TO n+16): NEXT n: GO TO 1830
1860 GO SUB 900: GO SUB 880: RET
URN
1870 LET px=p-4: IF px<=1 THEN
LET px=1
1880 LET py=p+4: IF py>=20 THEN
 LET py=20
1890 LET ox=o-4: IF ox<=1 THEN
LET ox=1
1900 LET oy=o+4: IF oy>=20 THEN
 LET oy=20
1910 RETURN
1920 LET d=d+d1: PRINT AT 21,7;d
;" ": BEEP .01,m
1930 IF d>=to THEN  FOR n=1 TO 6
0 STEP 10: BEEP .01,n: PRINT  OV
```

```
ER 1;AT 21,0;"PUNTOS:": NEXT n
1940 RETURN
1950 LET d=d-d1: PRINT AT 21,7;d
;"  "
1960 IF d<=0 THEN  GO TO 510
1970 BEEP .001,57: RETURN
1980 LET ca=INT (1+RND*LEN r$):
FOR n=7 TO 0 STEP -1: BEEP .01,a
: PRINT  FLASH 1;; PAPER n
;AT a1,b1;r$(ca): NEXT n
1990 LET w$=w$+r$(ca): PRINT  IN
K 7; PAPER 1;AT 0,0;w$: IF LEN w
$>=10 THEN  PRINT AT 0,0;w$: LET
 w$=""
2000 FOR n=2 TO 18: PRINT AT n,1
;"                              ":
NEXT n: GO SUB 480
2010 LET d1=125: GO SUB 1920: GO
 SUB 570
2020 IF d>=to THEN  BEEP 1,30: G
O SUB 2040
2030 RETURN
2040 IF en=0 THEN  PRINT  FLASH
1;AT 21,13;"y el enano...?": BEE
P 1,-20
2050 IF ro<7 THEN  PRINT  FLASH
1;AT 21,13;"y los 6 objetos?": B
EEP 1,-20
2060 IF en=0 OR ro<7 THEN  PRINT
 AT 21,13;"                  ": RE
TURN
2070 PRINT AT 1,3;"DESEAS VER LA
S CLAVES?"
2080 IF INKEY$="s" THEN  GO SUB
3700: GO TO 2110
2090 IF INKEY$="n" THEN  BEEP 2,
-30: GO TO 2110
2100 GO TO 2080
```




```
2110 LET a2=INT (1+RND*20): LET
b2=INT (1+RND*20): LET a(a2,b2)=
20
2120 PRINT AT 1,3;"
         ";AT 20,0;"
           ": POKE 23658,8: PRI
NT  FLASH 1;AT 0,0;w$; FLASH 0;"
 DELETREA LA CLAVE ";AT 21,0;"DE
LETE PARA BORRAR TODO    "
2130 LET z$="": PRINT AT 19,3;"T
IEMPO:      ": FOR n=1 TO 300: N
EXT n: FOR n=200 TO 1 STEP -1: B
EEP .0001,60: PRINT AT 19,11;n;"
 "
2140 IF INKEY$=CHR$ 13 OR INKEY$
=CHR$ 12 THEN  PRINT AT 20,13;"
          ": LET z$="": NEXT n:
POKE 23658,0
2150 IF INKEY$<>"" THEN  LET z$=
z$+INKEY$: BEEP 1,30
2160 PRINT AT 20,13; BRIGHT 1; I
NK 0;z$: IF z$=r$ THEN  POKE 236
58,0: GO TO 2190
2170 NEXT n: POKE 23658,0
2180 PRINT AT 21,0;"
          ": GO TO 2200
2190 IF z$=r$ THEN  PRINT AT 21,
12; FLASH 1; BRIGHT 1; INK 1;"**
 CORRECTO **": GO SUB 2930: PRIN
T AT 21,12;"                ": RET
URN
2200 IF z$<>r$ THEN  LET to=to+d
: LET nm=0: LET d1=500: GO SUB 1
950: PRINT AT 21,13;r$;"      ":
 FOR n=1 TO 300: NEXT n: LET w$=
"": PRINT  FLASH 1;AT 21,13;"NUE
VA CLAVE    ": GO SUB 3710: GO S
UB 1530: PRINT AT 21,10;"
          ": GO SUB 750: GO SU
B 480: RETURN
2210 REM  Coge objeto
2220 IF a(p,o)>=9 AND a(p,o)<40
THEN  LET vb=1: GO SUB 1980: PRI
NT AT a1,b1;" ": LET a1=0: LET b
1=0: LET a(p,o)=INT (1+RND*6): I
F a(p,o)=1 THEN  FOR n=1 TO 19:
BEEP .01,40: PRINT  OVER 1;AT 21
,30;"T": NEXT n: RETURN
2230 IF vb=1 THEN  LET vb=0: RET
URN
2240 IF a(p,o)=40 AND t=1 THEN
LET en=1
```


N.º 4
PRIMERA REVISTA EN CASSETTE
ABRIL-MAYO 85
SPECTRUM
COMPUTING
PARA 16K Y 48K
3 D
Añada una nueva dimensión a su SPECTRUM.
Acción en tres dimensiones.
Busque y destruya la flota de ataque estelar.
Piérdase en nuestro laberinto en tres dimensiones en código máquina.
Defensa
Su oportunidad para venir en defensa del planeta.
875 Ptas.
Más sobre la programación de dibujos animados.
Clocks up
¿Sabe generar un reloj digital en su SPECTRUM?
Y MUCHO MAS
BIENVENIDO A
SPECTRUM
COMPUTING
LABERINTO EN TRES DIMENSIONES
GRAFICOS
WIZARD
RELOJ
DEFENSA
DIBUJOS ANIMADOS
ATAQUE ESTELAR
AGENDA TELEFONICA
SKI
PUZLE
LA SERPIENTE
MAS DE 150.000 PTAS. EN PREMIOS. BASES EN EL INTERIOR
Solicítela a: INFODIS C/ Bravo Murillo, 377 - 5.º A - 28020 MADRID
CUPON DE PEDIDO
Nº 4
Sí, envíenme al precio de 875 ptas.... ejs. de SPECTRUM COMPUTING
El importe lo abonaré: Contra reembolso □ Con mi tarjeta de crédito □
Adjunto cheque □ American Express □ Visa □ Interbank □
Número de mi tarjeta
Fecha de caducidad
NOMBRE
DIRECCION
CIUDAD
PROVINCIA
Sin gastos de envío

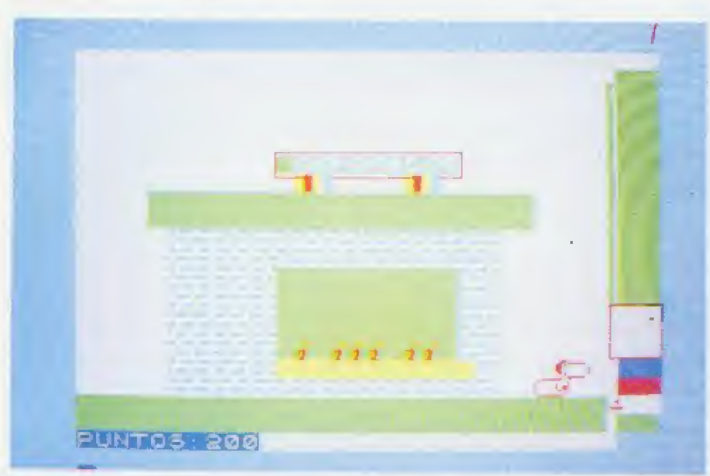
Programas

```
2250 IF a(p,o)=1 THEN GO SUB 23
80: RETURN
2260 IF p=a3 AND o=b3 THEN  LET
en=1: LET t=1: LET a3=0: LET b3=
0: GO TO 2360
2270 PRINT AT a1,b1;"J": BEEP .0
1,40+p: BEEP .01,40+o: BEEP .01,
50: PRINT AT a1,b1;" "
2280 IF a(p,o)>1 AND a(p,o)<>40
THEN  FOR n=a1+1 TO 18: PRINT  B
RIGHT 1; INK 4; PAPER 1;AT n,b1-
1;"SSS": BEEP .001,n+20: PRINT A
T n,b1-1;"   ": NEXT n: GO SUB 1
340
2290 IF a(p,o)=40 THEN  GO SUB 9
20: GO TO 2340
2300 FOR n=1 TO 6: IF o$=p$(n) O
R o$="J" THEN  GO TO 2330
2310 NEXT n
2320 LET p$(ro)=o$: LET ro=ro+1:
 PRINT AT co,30;o$: LET co=co+2
2330 IF o$="J" THEN  LET a1=0: L
ET b1=0: LET d1=ti*500: GO SUB 1
920: LET a(p,o)=40: RETURN
2340 LET a(p,o)=0
2350 LET a1=0: LET b1=0
2360 LET d1=1000: GO SUB 1920
2370 RETURN
2380 FOR n=w TO w+40
2390 IF n>=168 THEN  BEEP .1,45:
 GO TO 2410
2400 BEEP .0001,60: PLOT  INK 8;
 PAPER 8;232,n: DRAW  INK 8; PAP
ER 8;2,0: LET w=w+1: NEXT n: BEE
P .1,45
2410 LET w=w-1: PRINT  PAPER 4;A
T a1,b1;" ": LET a1=0: LET b1=0:
 RETURN
2420 REM  PLAYA
2430 DATA 63,127,255,255,255,255
,127,63,255,222,190,190,190,255,
204,240,30,33,76,77,79,37,20,30,
0,252,254,255,255,255,254,252,25
5,255,218,37,255,255,255,255,64,
10,88,8,73,56,125,48,0,73,232,72
,8,57,123,49,0,146,23,18,16,156,
222,140,0,24,24,32,192,0,0,0,32,
48,56,60,62,48,255,126
2440 DATA 255,222,190,166,255,25
4,252,240,56,120,184,144,126,57,
41,108,56,120,57,17,126,184,168,
108,0,96,96,16,16,224,0,0
2450 RESTORE 2430: FOR n=-29 TO
58: BORDER 0: BORDER 4: BORDER 2
: READ j: POKE USR "a"+n+29,j: B
EEP .001,n: BEEP .01,n-2: BEEP .
01,30-n: NEXT n
2460 FOR n=0 TO 23: BORDER 7: BO
RDER 4: BORDER 1: READ j: POKE U
SR "o"+n,j: BEEP .01,n: BEEP .01
,n-2: BEEP .01,60-n: BEEP .01,30
-n: NEXT n
2470 BORDER 5: PAPER 6: INK 1: B
RIGHT 1: CLS
2480 FOR n=0 TO 15: PRINT AT n,0
; PAPER 5;"
            ": NEXT n
2490 PLOT  INK 7;100,48: FOR n=1
 TO 11: DRAW  INK 7;14,-3+RND*15
,-2.5: NEXT n
2500 FOR n=0 TO 47: PLOT  INK 1;
80,47: DRAW  INK 1;175,-n: NEXT
n
2510 FOR n=11 TO 12: PRINT AT n,
4; PAPER 4; INK 2;" █████": NEXT
n: PRINT AT 10,5; INK 2; PAPER 4
;" ███"
2520 FOR n=12 TO 19: PRINT AT n,
1; INK 2; BRIGHT 0; PAPER 4;"███
"; BRIGHT 1;"█████": NEXT n
2530 FOR n=77 TO 100: PLOT  INK
2;77-n,n: DRAW  INK 2;30,0: NEXT
 n
2540 FOR n=100 TO 77 STEP -1: PL
OT  INK 2;147-n,-n: DRAW  INK 2;
14,0: NEXT n
2550 FOR n=13 TO 19: PRINT  INK
7;AT n,5;"███ ": NEXT n
2560 PRINT AT 18,26; INK 1; PAPE
R 5;"EEE";AT 17,31;"E";AT 18,30;
"E";AT 17,20;"EEEEE"
2570 FOR n=1 TO 5: PLOT RND*245,
120+RND*50: DRAW 5,-5,-2+RND*2:
DRAW 5,5,-2+RND*2: NEXT n
2580 PRINT AT 19,12;" CD";AT 20,
10;"T ABQ U   P": PAUSE 100
2590 LET es1=0: LET es=0
2600 RESTORE 2680: FOR n=31 TO 1
0 STEP -1: PRINT AT 20,20;" ": P
RINT  OVER 1; INK 8; PAPER 5;AT
15,n;"J": READ j: PRINT AT 19,16
;"GGH": PRINT AT 20,21;"O": BEEP
 n/500,j+es: PRINT AT 20,13;"KQ"
: PRINT AT 20,21;" ": PRINT AT 1
9,16;"FHG": READ j: BEEP .05,j+e
```

```
s: PRINT AT 19,16;"HFG": PRINT A
T 20,20;"P"; PAPER 5; INK 8; OVE
R 1;AT 15,n;"J"
2610 PRINT AT 20,14;"I": PRINT A
T 20,13;"B":
2620 IF INKEY$="s" THEN  RUN
2630 IF INKEY$="n" THEN  GO SUB
3780: GO TO 3560
2640 PRINT AT 20,20;"P": BEEP .0
01,60
2650 NEXT n: LET es1=es1+1: IF e
s1>=2 THEN  LET es=6
2660 IF es1=5 THEN  LET es=-12:
PRINT  PAPER 1; INK 6; FLASH 1;#
1;"OTRA PARTIDA? (S/N)"
2670 GO TO 2600
2680 DATA 0,5,0,7,0,8,0,10,0,12,
0,10,0,8,0,7,0,5,17,19,20,19,18,
17,17,19,20,19,17,15,15,17,19,17
,15,13,13,15,17,15,13,12,13,12
2690 FOR n=1 TO 13: PRINT AT n,2
9; BRIGHT 1; INK 4;" ███": NEXT n
2700 LET to=INT (10000+RND*20000
): PRINT  BRIGHT 1;AT 20,0;"A ";
to;" puntos"
2710 FOR n=17 TO 21: PRINT AT n,
29; BRIGHT 1; INK n-16;" ███": NE
XT n
2720 FOR n=64 TO 167: PLOT  INK
8; PAPER 8;232,n: DRAW  INK 8; P
APER 8;2,0: LET w=w+1: NEXT n: B
EEP .01,40
2730 PAUSE 200: RETURN
2740 FOR n=19 TO 20: PRINT AT n,
0; BRIGHT 1; INK 4;"██████████
███████████████": NEXT n
2750 FOR n=12 TO 18: PRINT AT n,
5; BRIGHT 1; INK 2;"EEEEEEEEEEEE
EE": NEXT n: PRINT AT 11,5; INK
2;"EEEEEEEEEEEEEE"
2760 FOR n=11 TO 18: PRINT INK 2
;AT n,1;"EEEE": NEXT n
2770 FOR n=83 TO 119: PLOT  INK
2;5+n-88,n: DRAW  INK 2;150-2*n+
176,0: NEXT n
2780 PRINT AT 13,6; BRIGHT 1; PA
PER 1; INK 4;"█▀▀▀▀█";AT 14,6;"█
█";AT 15,6;"█▄▄▄▄█"
2790 PRINT AT 13,2; PAPER 1; INK
4;"█▀█";AT 14,2;"█ █";AT 15,2;"█▄█
2800 FOR n=6 TO 9: PRINT AT n,14
; INVERSE 1; INK 3;"E";AT n,15;
BRIGHT 1;"EE": NEXT n
2810 PRINT  BRIGHT 1; PAPER 5; I
NK 4;AT 13,12;"█▀▀▀▀█": FOR n=14 T
O 18: PRINT AT n,12; PAPER 5; BR
IGHT 1; INK 4;"█    █": NEXT n
2820 FOR n=15 TO 18: PRINT AT n,
19; INVERSE 1; INK 2; BRIGHT 1;"
EEE": NEXT n
2830 FOR n=0 TO 30: PLOT  INK 2;
152,78: DRAW  INK 2;n,-22: NEXT
n
2840 PRINT AT 16,19; PAPER 1; IN
K 4; BRIGHT 1;"▀▀█";AT 17,19;"▄▄█"
2850 PRINT AT 2,2;"GUSANEZ HOUSE
": PLOT 85,120: DRAW 0,20: PLOT
85,135: DRAW 5,5: PLOT 85,135: D
RAW -5,5
2860 PRINT AT 5,3;"J": PLOT 37,1
20: DRAW 0,15: PLOT 37,131: DRAW
 -5,0: DRAW 17,0: DRAW -5,2: DRA
W 0,-4: DRAW 5,2
2870 IF nx=1 THEN  RETURN
2880 FOR n=7 TO 47 STEP 8: FOR m
=0 TO 25 STEP 2: PLOT  INK 1;110
+n+RND*20,128+RND*n: NEXT m: NEX
T n
2890 PRINT AT 0,0;"
                 ": GO SUB 880
: RETURN
2900 REM  Gusanez menos
2910 FOR m=1 TO 10: FOR n=232 TO
 234: BEEP .001,60: PLOT  OVER 1
; INK 8; PAPER 8;n,64: DRAW  OVE
R 1; INK 8; PAPER 8;0,103: NEXT
n: NEXT m
2920 GO TO 3550
2930 REM  Coche
2940 PRINT AT 0,0;"

                  "
2950 LET x1=4: LET x2=23: LET x3
=5: LET x4=15
2960 GO SUB 480: RESTORE 2990: F
OR n=0 TO 31: READ j: POKE USR "
a"+n,j: NEXT n
2970 FOR n=0 TO 31: READ j: POKE
 USR "f"+n,j: NEXT n
2980 FOR n=0 TO 7: READ j: POKE
USR "p"+n,j: NEXT n: GO SUB 480
2990 DATA 63,127,255,129,152,164
,103,24,241,254,254,130,50,74,20
4,48,252,59,255,112,112,52,212,1
9,192,252,194,193,1,1,2,252
```

Programas




```
3000 DATA 252,254,255,129,25,37,
230,24,143,127,127,65,76,82,51,1
2,63,220,255,14,14,44,43,200,3,6
3,67,131,128,128,64,63
3010 DATA 15,96,3,110,0,250,0,0
3020 LET ch=1
3030 PRINT AT 21,30; FLASH 1;"P"
3040 GO SUB 920: RETURN
3050 PRINT AT 1,5;"CAMBIO DE SIT
IO": FOR n=1 TO 19: GO SUB 480:
BORDER RND*7: BEEP .01,50: NEXT
n: BORDER 2
3060 FOR m=5 TO 0 STEP -2: LET d
1=500: GO SUB 1950: INK m: FOR n
=60 TO 133 STEP 8: PLOT 90+n,20+
n: DRAW  PAPER m; BRIGHT 1;55-(2
*n),0: DRAW  PAPER m; BRIGHT 1;0
,127-(2*n): DRAW  PAPER m; BRIGH
T 1;-55+(2*n),0: DRAW  PAPER m;
BRIGHT 1;0,-127+(2*n)
3070 NEXT n: INK m+2: FOR n=68 T
O 133 STEP 8: PLOT 90+n,20+n: DR
AW  PAPER m+2;55-(2*n),0: DRAW
PAPER m+2;0,127-(2*n): DRAW  PAP
ER m+2;-55+(2*n),0: DRAW  PAPER
m+2;0,-127+(2*n)
3080 NEXT n: NEXT m: INK 2
3090 LET p=INT (1+RND*20)
3100 LET o=INT (1+RND*20)
3110 LET d1=100: GO SUB 1950: GO
 SUB 750: GO SUB 480: RETURN
3120 REM  Movimiento enano
3130 IF en=1 THEN  RETURN
3140 LET nu=INT (RND*8): LET w1=
INT (RND*2): LET q=INT (RND*2)
3150 IF nu=0 THEN  LET a3=a3+1:
LET b3=b3+w1
3160 IF nu=1 THEN  LET a3=a3-q:
LET b3=b3-w1
3170 IF nu=2 THEN  LET a3=a3+q:
LET b3=b3-w1
3180 IF nu=3 THEN  LET a3=a3-q:
LET b3=b3+w1
3190 IF a3>=20 THEN  LET a3=2
```



```
3200 IF a3<=1 THEN  LET a3=20
3210 IF b3<=1 THEN  LET b3=20
3220 IF b3>=20 THEN  LET b3=1
3230 LET a(a3,b3)=-5
3240 RETURN
3250 REM  Dibujo kiosko
3260 LET b=24: FOR n=19 TO 20: P
RINT AT n,0; BRIGHT 1; PAPER 4;"
                                ":
NEXT n
3270>PRINT AT 10,5; INK 5;"EEEEE
EEEEEEEEEEEEE"
3280 FOR n=11 TO 18: PRINT INK 5
;AT n,5; BRIGHT 1;"EEEE"; BRIGHT
 0;"EEEEEEEEEEEEEE": NEXT n
3290 FOR n=12 TO 16: PRINT  PAPE
R 4;AT n,11;"             ": NEXT n
3300 PRINT  PAPER 4; INK 2;AT 16
,11;" T TTT TT "
3310>PRINT INK 6;AT 17,11; BRIGH
T 1;"█"; BRIGHT 0;"[redacted]"
3320 FOR n=6 TO 9: PRINT INK 4;A
T n,4; BRIGHT 1;"[redacted]"; BRIGHT 0
;"[redacted]": NEXT n
3330 PRINT AT 7,12; BRIGHT 1; PA
PER 6; INK 5;" █";AT 7,18;" █"
3340 PRINT  INK 4; BRIGHT 1;AT 6
,11;"█"; FLASH 1; BRIGHT 1;"REFR
ESCOS"
3350 PLOT 87,118: DRAW 82,0: DRA
W 0,11: DRAW -82,0: DRAW 0,-11
3360 PRINT AT 7,13; PAPER 6; INK
 5;" █";AT 7,19;" █"
3370 RETURN
3380 REM  Pantalla objeto
3390 IF a(p,o)=8 THEN  RETURN
3400 LET a1=INT (4+RND*14): LET
b1=INT (2+RND*25)
3410 FOR n=18 TO a1+1 STEP -1: P
RINT  INK 4; PAPER 1;AT n,b1;" S"
: NEXT n: PRINT AT a1+1,b1-1; OV
ER 1; BRIGHT 1; INK 4; PAPER 1;"
S S"
3420 RETURN
```

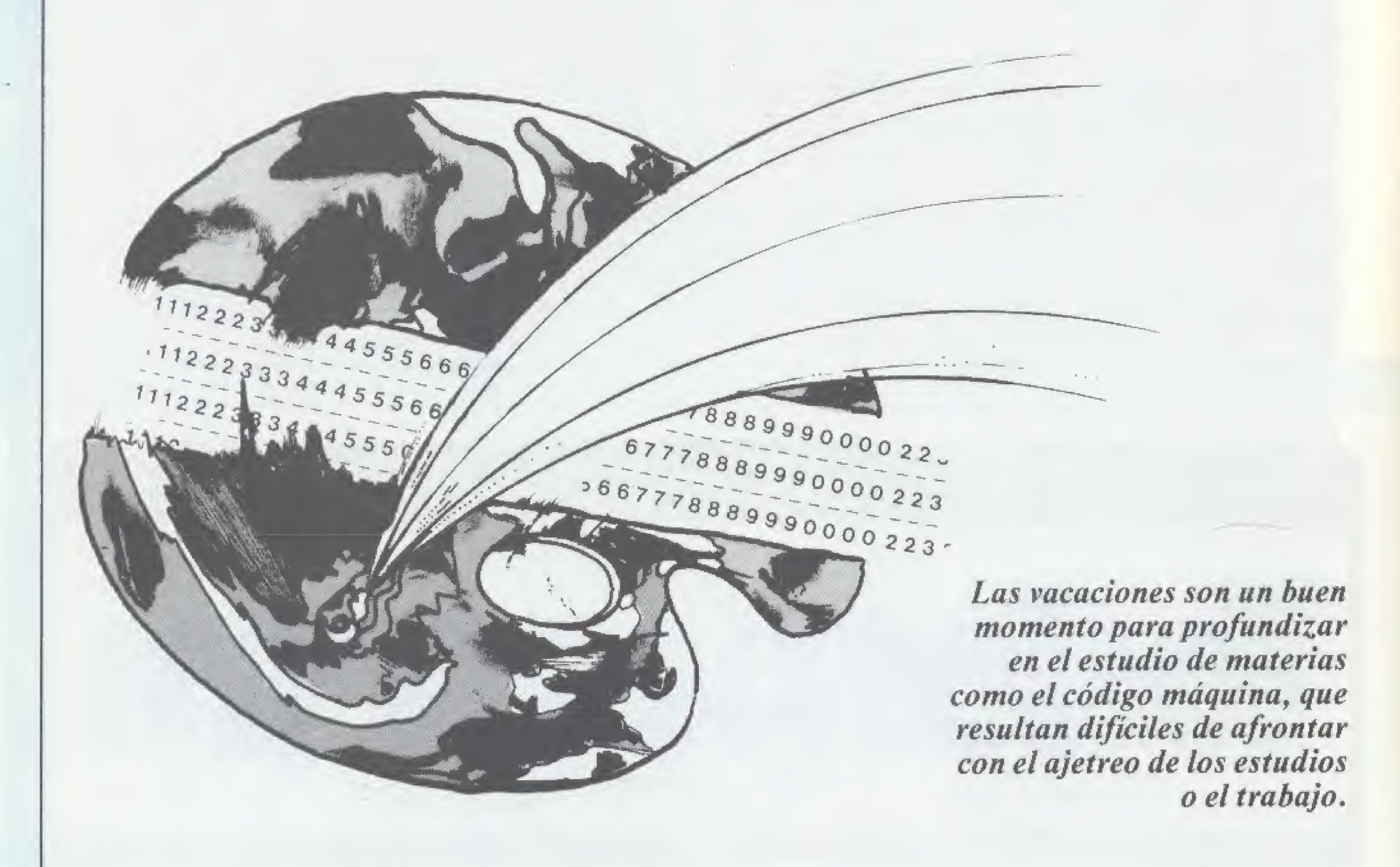
Programas

```
3430 REM  Pantalla especial
3440 IF ch=1 THEN  RETURN
3450 FOR n=1 TO 19: PRINT AT n,0
; BRIGHT 1; INK 5; PAPER 3;"J";A
T n,28;"J": NEXT n
3460 FOR n=0 TO 28: PRINT AT 1,n
; INK 5; PAPER 3; BRIGHT 1;"J";A
T 19,n;"J": NEXT n
3470 GO SUB 570
3480 GO SUB 990
3490 PRINT  INK 6; PAPER 1;AT 3,
13;"█GRANDES";AT 4,12;"█ALMACENE
S"
3500 LET a1=INT (5+RND*13): LET
b1=INT (2+RND*25): FOR m=1 TO 5:
 FOR n=0 TO 7: PRINT AT a1,b1; I
NK n;o$: BEEP .001,n+50: NEXT n:
 NEXT m
3510>FOR n=5 TO 18: PRINT PAPER
5; INK RND*7;AT n,1;"[][][][]";
PAPER 4;"[][][][][][][][][]": NE
XT n
3520 GO SUB 480
3530 GO SUB 480: LET en=0: GO SU
B 920
3540 RETURN
3550 REM  Final gusanez
3560 LET nx=1: PAPER 0: BORDER 0
: INK 5: CLS : GO SUB 2740
3570 FOR n=1 TO 50: PLOT  BRIGHT
 1; INK 7;RND*255,130+RND*45: NE
XT n: PLOT  INK 7;220,120: DRAW
 INK 7; BRIGHT 1;10,-30,4.3: DRA
W  INK 7; BRIGHT 1;-10,30,-2.5
3580 LET en=0: LET a=18: LET b=2
6: LET g$=" GF": LET h$="IH ": G
O SUB 490
3590 FOR n=26 TO 12 STEP -1: GO
SUB 490: LET b=n: GO SUB 490: PA
USE 10: NEXT n: PAUSE 30: GO SUB
 490
3600 PAUSE 50: PRINT  INVERSE 1;
 PAPER 4; INK 7; BRIGHT 1;AT 13,
6;"▛▀▀▀▀▜";AT 14,6;"▌    ▐";AT 15,6;
"▙▄▄▄▄▟"
3610 BEEP .001,30: PAUSE 75: PRI
NT  INVERSE 1; BRIGHT 1; PAPER 4
; INK 7;AT 13,2;"▛▀▜";AT 14,2;"▌ ▐
";AT 15,2;"▙▄▟"
3620 BEEP .001,30: PAUSE 100: PR
INT  INK 4;AT 13,6;"▛▀▀▀▀▜";AT 14,
6;"▌    ▐";AT 15,6;"▙▄▄▄▄▟"
3630 BEEP .001,30: PAUSE 250: PR
INT  INK 4;AT 13,2;"▛▀▜";AT 14,2;
"▌ ▐";AT 15,2;"▙▄▟": BEEP .001,30:
 PAUSE 250
3640 FOR n=1 TO 7: PRINT  INK n;
AT 21,1;"S PARA JUGAR  N para re
seteo": PAUSE 5: NEXT n
3650 FOR n=1 TO 6: BEEP .01,50
3660 IF INKEY$="s" THEN  LET no=
0: RUN
3670 IF INKEY$="n" THEN  RANDOMI
ZE USR 0
3680 NEXT n: PAUSE 5
3690 GO TO 3650
3700 PRINT AT 8,2;"CLAVES:": PAU
SE 100: RESTORE 3720: FOR n=1 TO
 35: READ j$: FOR m=7 TO 0 STEP
-1: PRINT  INK m;AT 4,3;j$;"
    ": NEXT m: FOR m=0 TO 7: PRIN
T  INK m;AT 4,3;j$;"         ": N
EXT m: BEEP .01,60: NEXT n: RETU
RN
3710 RESTORE 3720: LET no=INT (1
+RND*35): FOR n=1 TO no: READ j$
: NEXT n: LET r$=j$: RETURN
3720 DATA "TECLADO","ORDENADOR",
"PROGRAMA","LISTADO","TELEVISION
","CIRCUITO","BINARIO","DECIMAL"
,"PANTALLA","PERSONAL","ALEATORI
O","IMPRESORA","INTERFACE","BUS"
,"MATRIZ","BUCLE","VARIABLE","PA
NTALLA","SUBRUTINA","GUSANEZ","D
ISCO","MONITOR","RUTINA","DECISI
ON","GRABACION","SONIDO","RESET"
,"GRAFICO","ELECTRON","INTEGRADO
","CURSOR","BORDE","INVERSO","BO
RRAR","SINCLAIR"
3730 PRINT "****** LAS VACACIONE
S DE *******"''
3740>FOR w=0 TO 1: FOR m=0 TO 1:
 FOR n=7 TO 1 STEP -1: BEEP .01,
n: BORDER n: PRINT AT 2,0; INVER
SE w; INK n; BRIGHT m;"
ISANEZ  GL
": NEXT n
: NEXT m
3750 NEXT w
3760 PRINT ' INK 1;"* GUSANEZ SE
 MUEVE CON Z,X,P y L"': PRINT  F
LASH 1;"  SE PARA CON OTRA TECLA
```

```
"'': PRINT "* COGE LOS 6 OBJETOS
 DIFERENTES"'': PRINT "* BUSCA A
L ENANO Y CUIDADO CON    LOS GRA
NDES ALMACENES"'': PRINT "* CONV
IERTE A GUS EN UN COCHE     COGI
ENDO LA ULTIMA FLOR Y...": PRINT
 ;"  ACERTANDO LA CLAVE SECRETA"
''
3770 PRINT  INK 4;"* BEBE REFRES
COS POR EL CAMINO"'': PRINT  FLA
SH 1;"* TU META ES LLEGAR A LA P
LAYA"
3780 RESTORE 3790: FOR n=0 TO 16
7: BORDER 2: BORDER 4: BORDER 1:
: READ j: POKE USR "a"+n,j: NEXT
 n: RETURN
3790 DATA 63,64,128,128,128,128,
64,63,241,2,66,78,78,50,4,248,28
,59,116,112,112,52,20,19,0,252,2
,1,1,1,2,252,243,243,0,0,207,207
,0,0,252,2,1,1,1,1,2,252
3800 DATA 143,64,66,114,114,76,3
2,31,56,220,46,14,14,44,40,200,0
,63,64,128,128,128,64,63,153,90,
60,195,195,60,90,153,129,195,129
,129,129,255,231,231
3810 DATA 60,66,255,231,223,211,
231,255,192,192,128,188,254,195,
195,126,0,255,255,129,171,213,17
1,255,195,231,24,187,190,184,255
,24
3820 DATA 56,248,56,16,124,186,4
0,40,221,221,0,119,119,119,0,221
,170,85,170,85,170,85,170,85,255
,68,255,17,255,68,255,17,7,8,16,
120,88,88,48,120,126,129,173,255
,195,231,195,255
3830 SAVE "gusanez" LINE 1: BEEP
 5,10: VERIFY "": BEEP 1,10
9999 SAVE "gusanez" LINE 1: BEEP
 5,10: VERIFY "": BEEP 1,10
```

Programas (52)

Monitor de Código Máquina



Las vacaciones son un buen momento para profundizar en el estudio de materias como el código máquina, que resultan difíciles de afrontar con el ajetreo de los estudios o el trabajo.

Coincidiendo con el inicio de una serie de código máquina en nuestra revista, os queremos presentar una herramienta muy útil a la hora de cargar o depurar los programas que se realizarán en este curso.

Pablo Torrente nos ha enviado un trabajo bastante interesante: su programa permite listar en hexadecimal los contenidos de la memoria, cargar bytes en memoria, desensamblar programas, ejecutarlos de una vez o paso a paso. También dispone de comandos para activar y desactivar la impresora.

El programa trabaja en Spectrum de 16 ó 48 K, si bien es incompatible con el Interface I, ya que, al residir el código máquina el líneas REM, la relocalización que producen las variables del sistema extra del Interface impide una ejecución correcta.

El uso general de los comandos es, para los que necesitan una dirección, teclear ésta en hexadecimal y después la letra del comando. Para los restantes, basta introducir el comando para que surta efecto. El comando de carga en memoria utiliza los bytes que le siguen en la línea de entrada.

La introducción del programa no es un problema trivial, y conviene seguir las instrucciones que os damos al pie de la letra. Para comenzar, hay que teclear el programa en BASIC (número uno), sin olvidarse de la línea 1, muy importante. Después de finalizar la introducción, recomendamos salvar

a cassette el programa. Tecleando GO TO 9000, se ejecuta un programa en código máquina que convierte la línea 1 en línea 0, dándole una longitud de 800 bytes. El programa carga allí el código máquina del comando L (introducir código) y T (listar código). Esto permite utilizar esa parte del monitor, que ya funciona, para introducir, a partir de la localidad 5CD2, los bytes en hexadecimal correspondientes al listado 2.

Es importante no tratar de utilizar otros comandos que L o T hasta no haber finalizado la introducción del código. El comando T es importante, ya que permite verificar los bytes introducidos. Al acabar, basta salvar a cassette de nuevo el programa. Dispondremos, si la introducción del código máquina ha sido correcta, de nuestro monitor.

El monitor, de reducido tamaño, no desensambla los mnemónicos Z80, pero cumple con la tarea de separar las instrucciones correctamente, distinguiendo las de un byte de las de dos o tres, etc. El comando de ejecución paso a paso resulta satisfactorio, y el programa entero puede servir como ayuda, a la vez que su estudio proporciona conocimientos interesantes de código máquina.

Autor: Pablo Torrente ***16K***

Ejecución paso a paso Dirección 1602

```
1602-    C6 16
 AF=1600 BC=0000 DE=0000 HL=0000 IX=0000 IY=5C3A SP=FF56

1604-    6F
 AF=1600 BC=0000 DE=0000 HL=0016 IX=0000 IY=5C3A SP=FF56

1605-    26 5C
 AF=1600 BC=0000 DE=0000 HL=5C16 IX=0000 IY=5C3A SP=FF56

1607-    5E
 AF=1600 BC=0000 DE=0001 HL=5C16 IX=0000 IY=5C3A SP=FF56

1608-    23
 AF=1600 BC=0000 DE=0001 HL=5C17 IX=0000 IY=5C3A SP=FF56

1609-    56
 AF=1600 BC=0000 DE=0001 HL=5C17 IX=0000 IY=5C3A SP=FF56

160A-    7A
 AF=0000 BC=0000 DE=0001 HL=5C17 IX=0000 IY=5C3A SP=FF56

160B-    B3
 AF=0100 BC=0000 DE=0001 HL=5C17 IX=0000 IY=5C3A SP=FF56

160C-    20 02
 AF=0100 BC=0000 DE=0001 HL=5C17 IX=0000 IY=5C3A SP=FF56

1610-    1B
 AF=0100 BC=0000 DE=0000 HL=5C17 IX=0000 IY=5C3A SP=FF56

1611-    2A 4F 5C
 AF=0100 BC=0000 DE=0000 HL=5CB6 IX=0000 IY=5C3A SP=FF56

1614-    19
 AF=0108 BC=0000 DE=0000 HL=5CB6 IX=0000 IY=5C3A SP=FF56

1615-    22 51 5C
 AF=0108 BC=0000 DE=0000 HL=5CB6 IX=0000 IY=5C3A SP=FF56
```

Desensamblado Dirección 0

```
0000-    F3
0001-    AF
0002-    11 FF FF
0005-    C3 CB 11
0008-    2A 5D 5C
000B-    22 5F 5C
000E-    18 43
0010-    C3 F2 15
0013-    FF
0014-    FF
0015-    FF
0016-    FF
0017-    FF
0018-    2A 5D 5C
001B-    7E
001C-    CD 7D 00
001F-    D0
0020-    CD 74 00
0023-    18 F7
0025-    FF
```

Programas

(52)

```
   1 REM *
   2 REM  MONITOR CODIGO MAQUINA
   3 REM   P.TORRENTE    1985
   4 REM      instrucciones
   5 REM 1)-Cada comando puede
ir precedido (opcionalmente) por
una direccion en  hexadecimal
   6 REM 2)-Comandos :
   7 REM L=cargar en memoria los
          bytes dados tras "L"
        M=desensambla en hex
        R=corre un programa

   8 REM S=desensambla y ejecuta
          paso a paso un progra
          ma en codigo maquina,
          mostrando reg.CPU
          (s=sig instr.
          SPACE=volver al menu)

   9  REM T=visualizar memoria
        Y=retorno a BASIC
        P1=impresora
        P0=pantalla
  20 BORDER 1: PAPER 1: INK 7: C
LEAR
  30 LET pc=0
 100 POKE 23658,8: POKE 23617,25
2: INPUT  LINE c$
 200 IF c$="" THEN  GO TO 100
 300 LET l$=c$(1): LET c$=c$(2 T
O )
 400 IF l$=" " THEN  GO TO 200
 500 IF l$="L" THEN  GO TO 1300
 600 IF l$="M" THEN  GO TO 2000
 650 IF l$="P" THEN  GO TO 2620
 700 IF l$="R" THEN  GO TO 2700
 800 IF l$="S" THEN  GO TO 3000
 900 IF l$="T" THEN  GO TO 3400
 910 IF l$="Y" THEN  STOP
1000 LET c$=l$+c$
1100 GO SUB 3600
1105 LET pc=h
1110 LET l$=CHR$ (code)
1200 GO TO 400
1300 REM ******CARGA BYTES******
1400 IF c$="" THEN  GO TO 100
1600 GO SUB 3600
1700 POKE pc,h
1800 LET pc=pc+1
1900 GO TO 1400
2000 REM ******** MONITOR ******
2100 FOR L=1 TO 20
2200 POKE 23809,pc-256*INT (pc/2
56): POKE 23810, INT (pc/256)
2300 RANDOMIZE USR 24275
2400 LET pc=pc+PEEK 23813
2500 NEXT L
2600 GO TO 200
2620 REM ****IMPRES./PANTALLA***
2630  GO SUB 3600
2640  POKE 24527,H+2: POKE 24276
,H+2
2650 GO TO 200
2700 REM **********RUN**********
2800 RANDOMIZE USR pc
2900 GO TO 200
3000 REM *********STEP**********
3100 POKE 23809,pc-256*INT (pc/2
56): POKE 23810, INT (pc/256)
3200 RANDOMIZE USR 23762
3210 LET pc=256*PEEK 23810+PEEK
23809
3300 GO TO 200
3400 REM ******PRINT BYTES******
3410 FOR I=1 TO 20
3500 POKE 23809,pc-256*INT (pc/2
56): POKE 23810, INT (pc/256)
3510  RANDOMIZE USR 24526
3520 LET pc=pc+8
3530 NEXT I
3540 GO TO 200
3600 REM *****HEX-DEC*********
3700 IF c$="" THEN  RETURN
3800 LET l$=c$(1): LET c$=c$(2 T
O )
3900 IF l$=" " THEN  GO TO 3700
4000 LET h=0
4100 LET code=CODE l$
4200 IF code<48 OR code>70 THEN
 RETURN
4300 IF code>57 AND code<65 THEN
 RETURN
4400 LET h=16*h+(code-48-7*(code
>64))
4500 IF c$="" THEN  RETURN
4700 LET l$=c$(1): LET c$=c$(2 T
O ): GO TO 4100
6000 SAVE "monitor" LINE 1
6789 RANDOMIZE 6789
6790 FOR l=23755 TO PEEK 23627+2
56*PEEK 23628 STEP 4
6795 IF PEEK (l-1)=13 AND PEEK l
=PEEK 23671 AND PEEK (l+1)=PEEK
23670 THEN  RANDOMIZE l: POKE 23
627,PEEK 23670: POKE 23628,PEEK
23671: PRINT ; FLASH 1'
"  Pulsa una tecla para seguir
": PAUSE 0: GO TO 20
6799 LET l=l+PEEK (l+2)+256*PEEK
 (l+3): NEXT l
9000 REM ***CARGADOR INICIAL***
9001 CLS
9002 PRINT '                     "
  Espera unos segundos hasta que
aparezca el cursor"
9003 PRINT '                     "
  Para cargar los bytes del lis-
tado hexadecimal adjunto,teclea
5CD2 L  seguido de los primeros
bytes.  Para  cargar sucesivos
bytes basta teclear L antes de
cada bloque. Se pueden verificar
los bytes mediante el comando T"
9004 PRINT '                     "
  Cuando termines de cargar los
 bytes puedes grabar el programa
 mediante GOTO 6000, ya que las
 lineas 6789-9918 habran desapa-
 recido y el codigo maquina se
 almacena en la linea 0"
9010 RESTORE 9800
9018 FOR i=0 TO 9
9020 READ b: POKE 40960+i,b
9022 NEXT i
9030 RANDOMIZE USR 40960
9032 POKE 23756,0: POKE 23757,35
9034 POKE 23758,3: POKE 23760,13
9036 POKE 23761,128
9040 FOR i=0 TO 59
9045 READ b: POKE 24501+i,b
9050 NEXT i
9060 GO TO 6789
9800 DATA 33,209,92,1,32,3,205,8
5,22,201
9900 DATA 87,31,31,31,31,205,194
9902 DATA 95,122,205,194,95,201,
230
9904 DATA 15,198,48,254,58,56,2
9906 DATA 198,7,215,201
9910 DATA 62,2,205,1,22,62,13
9912 DATA 215,42,1,93,124,205,18
1
9914 DATA 95,125,205,181,95,62,4
5
9916 DATA 215,6,8,62,32,215,126
9918 DATA 205,181,95,35,16,246,2
01
```

```
5CD2- 21 16 00 39 F9 3E FD DB
5CDA- FE CB 4F 20 06 18 57 FB
5CE2- CD F8 5C 3E 7F DB FE CB
5CEA- 47 20 EA 21 03 13 E5 ED
5CF2- 73 3D 5C C3 76 1B 11 00
5CFA- F0 13 7B B2 20 FB C9 FA
5D02- 5D 00 00 01 00 00 00 00
5D0A- 00 00 00 00 00 00 3A 5C
5D12- A3 00 00 00 00 00 00 3E
5D1A- 00 00 3A 5C 58 FF 41 46
5D22- 42 43 44 45 48 4C 49 58
5D2A- 49 59 53 50 00 00 00 00
5D32- C3 CA 5D C3 D0 5D CD D3
5D3A- 5E 21 2E 5D 11 AD 5D 01
5D42- 0A 00 ED B0 2A 01 5D 3A
5D4A- 05 5D 4F 7E FE CD CA 19
5D52- 5E FE C9 CA F7 5D FE E9
5D5A- CA 14 5E FE C3 CA 20 5E
5D62- FE DD 28 04 FE FD 20 11
5D6A- 23 5F 7E FE E9 20 07 BB
5D72- DA 0B 5E C3 02 5E 2B 18
5D7A- 1F E6 F7 FE 10 28 20 E6
5D82- E7 FE 20 28 1A E6 C7 FE
5D8A- C7 28 5E E6 06 FE 06 28
5D92- 07 7E E6 C1 FE C0 28 1D
5D9A- 11 AD 5D ED B0 18 09 11
5DA2- AD 5D 7E ED A0 EB 36 05
5DAA- CD 63 5E 00 00 00 00 C3
5DB2- CA 5D C3 D0 5D 7E E6 FB
5DBA- F6 02 32 C5 5D ED 5B 12
5DC2- 5D D5 F1 DA DE 5D 18 33
5DCA- CD 98 5E C3 FD 5D CD 98
5DD2- 5E 2A 01 5D 23 7E CD A1
5DDA- 5F C3 FA 5D 7E E6 06 CB
5DE2- 2F 28 12 FE 02 28 30 18
5DEA- 35 AE F5 CD 9E 5F F1 E5
5DF2- 6F AF 67 18 32 E1 18 2F
5DFA- 22 01 5D CD 9E 5F 18 27
5E02- CD 63 5E DD 22 01 5D 18
5E0A- 21 CD 63 5E FD 22 01 5D
5E12- 18 18 CD 63 5E 18 10 01
5E1A- 03 00 CD 9E 5F E5 2A 01
5E22- 5D 08 09 56 2B 5E EB 22
5E2A- 01 5D 1E 07 ED 73 1E 5D
5E32- 3E 0D D7 DD 21 12 5D 21
5E3A- 20 5D 3E 20 D7 06 02 7E
5E42- D7 23 10 FB 3E 3D D7 DD
5E4A- 7E 01 CD B5 5F DD 7E 00
5E52- CD B5 5F DD 23 DD 23 1D
5E5A- C2 3C 5E 3E 0D D7 C3 E1
5E62- 5C 2A 12 5D E5 F1 08 2A
5E6A- 06 5D E5 F1 08 ED 4B 14
5E72- 5D ED 5B 16 5D 2A 18 5D
5E7A- DD 2A 1A 5D FD 2A 1C 5D
5E82- D9 ED 4B 08 5D ED 5B 0A
5E8A- 5D 2A 0C 5D DD 2A 0E 5D
5E92- FD 2A 10 5D D9 C9 ED 43
5E9A- 14 5D F5 C1 ED 43 12 5D
5EA2- ED 53 16 5D 22 18 5D DD
5EAA- 22 1A 5D FD 22 1C 5D D9
5EB2- 08 ED 43 08 5D F5 C1 ED
5EBA- 43 06 5D ED 53 0A 5D 22
5EC2- 0C 5D DD 22 0E 5D FD 22
5ECA- 10 5D D9 08 C9 00 00 00
5ED2- 3E 3E 03 CD 01 16 3E 0D
5EDA- D7 06 02 21 01 5D 23 7E
5EE2- CD B5 5F 2B 10 F9 3E 2D
5EEA- D7 01 04 00 11 CF 5E 2A
5EF2- 01 5D ED B0 3A CF 5E 21
5EFA- 4F 5F 0E 03 ED B1 28 2D
5F02- 1E 03 0E 1C ED B1 28 08
5F0A- 1D 0E 1A ED B1 28 01 1D
```

```
5F12- 06 03 CD AF 5F 43 21 CF
5F1A- 5E 7E CD 85 5F 23 3E 20
5F22- D7 10 F6 06 09 CD AF 5F
5F2A- 7B 32 05 5D C9 3A D0 5E
5F32- 21 88 5F 1E 04 0E 0B ED
5F3A- B1 28 D5 1D 1D 0E 0B ED
5F42- B1 28 CD 3A CF 5E FE ED
```

```
5F4A- 28 C6 1C 18 C3 DD ED FD
5F52- 01 11 21 22 2A 31 32 3A
5F5A- C2 C3 C4 CA CC CD D2 D4
5F62- DA DC DD E2 E4 EA EC F2
5F6A- F4 FA FC FD 06 0E 10 16
5F72- 18 1E 20 26 28 2E 30 36
5F7A- 38 3E C6 CB CE D3 D6 DB
```

```
5F82- DE E6 ED EE F6 FE 21 22
5F8A- 2A 36 43 4B 53 5B 73 7B
5F92- CB 09 19 23 29 2B 39 E1
5F9A- E3 E5 E9 F9 3A 05 5D 2A
5FA2- 01 5D FE 80 38 01 25 85
5FAA- 6F 30 01 24 C9 3E 20 D7
5EB2- 10 FB C9
```

```
L
00010        ORG #5CD2
00020 ;
00030   ENT
00040        LD HL,#0016
00050        ADD HL,SP
00060        LD SP,HL
00070 MON    LD A,#FD
00080        IN A,(#FE)
00090        BIT 1,A
00100        JR NZ,NSTEP
00110        JR STEP
00120 NEWIN  EI
00130        CALL PSE
00140 NSTEP  LD A,#7F
00150        IN A,(#FE)
00160        BIT 0,A
00170        JR NZ,MON
00180        LD HL,#1303
00190        PUSH HL
00200        LD (#5C3D),SP
00210        JP #1876
00220 PSE    LD DE,#F000
00230 PS1    INC DE
00240        LD A,E
00250        OR D
00260        JR NZ,PS1
00270        RET
00280 PC     DEFW #7066
00290 RTN    DEFW 00
00300 LON    DEFB 0
00310 AF2    DEFW 00
00320 BC2    DEFW 00
00330 DE2    DEFW 00
00340 HL2    DEFW 00
00350 IX2    DEFW 00
00360 IY2    DEFW #5C3A
00370 AF1    DEFW 00
00380 BC1    DEFW 00
00390 DE1    DEFW 00
00400 HL1    DEFW 00
00410 IX1    DEFW 00
00420 IY1    DEFW #5C3A
00430 SPT    DEFW 00
00440 LBLS DEFM /AFBCDEHLIXIYSP/
00450 IBLQ   DEFW 0,0
00460        JP NJPRL
00470        JP JPREL
00480 STEP   CALL INST
00490        LD HL,IBLQ
00500        LD DE,XQAR
00510        LD BC,#000A
00520        LDIR
00530        LD HL,(PC)
00540        LD A,(LON)
00550        LD C,A
00560        LD A,(HL)
00570        CP #CD
00580        JP Z,CALLS
00590        CP #C9
00600        JP Z,RETS
00610        CP #E9
00620        JP Z,JPHLS
00630        CP #C3
00640        JP Z,XJUMP
00650        CP #DD
00660        JR Z,JP1
00670        CP #FD
00680        JR NZ,NXT
00690 JP1    INC HL
00700        LD E,A
00710        LD A,(HL)
00720        CP #E9
00730        JR NZ,JP2
00740        CP E
00750        JP C,JPIYS
00760        JP JPIXS
00770 JP2    DEC HL
00780        JR NO
00790 NXT    AND #F7
00800        CP #10
00810        JR Z,JRS
00820        AND #E7
00830        CP #20
00840        JR Z,JRS
00850        AND #C7
00860        CP #C7
00870        JR Z,RSTS
00880        AND #06
00890        CP #06
00900        JR Z,NO
00910        LD A,(HL)
00920        AND #C1
00930        CP #C0
00940        JR Z,COND
00950 NO     LD DE,XQAR
00960        LDIR
00970        JR XQINS
00980 JRS    LD DE,XQAR
00990        LD A,(HL)
01000        LDI
01010        EX DE,HL
01020        LD (HL),5
01030 XQINS  CALL LREG
01040 XQAR   DEFW 0,0,0,0,0
01050 COND   LD A,(HL)
01060        AND #F8
01070        OR #02
01080        LD (JMPC),A
01090        LD DE,(AF1)
01100        PUSH DE
```




```
        POP AF
JMPC    JP TRUE;↑ JP XX TRUE
FALS    JR INCP3
NJPRL   CALL SREG
        JP INCP3
JPREL   CALL SREG
        LD HL,(PC)
        INC HL
        LD A,(HL)
        CALL ADPC3
        JP INCP2
TRUE    LD A,(HL)
        AND  6
        SRA A
        JR Z,RETS
        CP 2
        JR Z,CALLS
        JR XJUMP
RSTS    XOR (HL)
        PUSH AF
        CALL ADPC2
        POP AF
        PUSH HL
        LD L,A
        XOR A
        LD H,A
        JR JUMPS
RETS    POP HL
        JR JUMPS
INCP2   LD (PC),HL
INCP3   CALL ADPC2
        JR JUMPS
JPIXS   CALL LREG
        LD (PC),IX
        JR JPEND
JPIYS   CALL LREG
        LD (PC),IY
        JR JPEND
JPHLS   CALL LREG
        JR JUMPS
CALLS   LD BC,0003
        CALL ADPC2
        PUSH HL
XJUMP   LD HL,(PC)
        DEC BC
        ADD HL,BC
        LD D,(HL)
        DEC HL
        LD E,(HL)
        EX DE,HL
JUMPS   LD (PC),HL
JPEND   LD E,7
        LD (SPT),SP
        LD A,#0D
        RST #10
        LD IX,AF1
        LD HL,LBLS
DSP1    LD A,#20
        RST #10
        LD B,2
DSP2    LD A,(HL)
        RST #10
        INC HL
        DJNZ DSP2
        LD A,"=
        RST #10
        LD A,(IX+1)
        CALL PRINT
        LD A,(IX+0)
        CALL PRINT
        INC IX
        INC IX
        DEC E
        JP NZ,DSP1
DSEND   LD A,#D
        RST #10
        JP NEWIN
LREG    LD HL,(AF1)
        PUSH HL
        POP AF
        EX AF,AF'
        LD HL,(AF2)
        PUSH HL
        POP AF
        EX AF,AF'
        LD BC,(BC1)
        LD DE,(DE1)
        LD HL,(HL1)
        LD IX,(IX1)
        LD IY,(IY1)
        EXX
        LD BC,(BC2)
        LD DE,(DE2)
        LD HL,(HL2)
        LD IX,(IX2)
        LD IY,(IY2)
        EXX
        RET
SREG    LD (BC1),BC
        PUSH AF
        POP BC
        LD (AF1),BC
        LD (DE1),DE
        LD (HL1),HL
        LD (IX1),IX
        LD (IY1),IY
        EXX
        EX AF ,AF'
        LD (BC2),BC
        PUSH AF
        POP BC
        LD (AF2),BC
        LD (DE2),DE
        LD (HL2),HL
        LD (IX2),IX
        LD (IY2),IY
        EXX
        EX AF,AF'
        RET
;
FCODE   DEFB 0
SCODE   DEFB 0
        DEFW 0
INST    LD A,2
        CALL #1601
        LD A,#0D
        RST #10
        LD B,2
        LD HL, PC
        INC HL
LOOP    LD A,(HL)
```

```
02420        CALL PRINT
02430        DEC HL
02440        DJNZ LOOP
02450        LD A,"-
02460        RST #10
02470        LD BC,4
02480        LD DE,FCODE
02490        LD HL,(PC)
02500        LDIR
02510        LD A,(FCODE)
02520        LD HL,CODES
02530        LD C,3
02540        CPIR
02550        JR Z,CHECK
02560 LOOP2  LD E,3
02570        LD C,#1C
02580        CPIR
02590        JR Z,P4
02600        DEC E
02610        LD C,#1A
02620        CPIR
02630        JR Z,P4
02640        DEC E
02650 P4     LD B,3
02660        CALL PRSPC
02670        LD B,E
02680        LD HL,FCODE
02690 LOOP3  LD A,(HL)
02700        CALL PRINT
02710        INC HL
02720        LD A,#20
02730        RST #10
02740        DJNZ LOOP3
02750        LD B,9
02760        CALL PRSPC
02770        LD A,E
02780        LD (LON),A
02790        RET
02800 CHECK  LD A,(SCODE)
02810        LD HL,CODE4
02820        LD E,4
02830        LD C,#0B
02840        CPIR
02850        JR Z,P4
02860        DEC E
02870        DEC E
02880        LD C,#B
02890        CPIR
02900        JR Z,P4
02910        LD A,(FCODE)
02920        CP #ED
02930        JR Z,P4
02940        INC E
02950        JR P4
02960 ;
02970 CODES DEFB #DD,#ED,#FD
02980 CODE3 DEFB #01,#11,#21,#22
02990 DEFB #2A,#31,#32,#3A,#C2
03000 DEFB #C3,#C4,#CA,#CC,#CD
03010 DEFB #D2,#D4
03020 DEFB #DA,#DC,#DD,#E2,#E4
03030 DEFB #EA,#EC,#F2,#F4,#FA
03040 DEFB #FC,#FD
03050 CODE2 DEFB 6,#E,#10,#16
03060 DEFB #18,#1E,#20,#26,#28
03070 DEFB #2E,#30,#36,#38,#3E
03080 DEFB #C6,#CB,#CE,#D3,#D6
03090 DEFB #DB,#DE,#E6,#ED,#EE
03100 DEFB #F6,#FE
03110 ;
03120 CODE4 DEFB #21,#22,#2A,#36
03130 DEFB #43,#4B,#53,#5B
03140 DEFB #73,#7B,#CB
03150 COD2A DEFB  9,#19,#23,#29
03160 DEFB #2B,#39,#E1,#E3,#E5
03170 DEFB #E9,#F9
03180 ;
03190 ;
03200 ADPC2 LD A,(LON)
03210 ADPC3 LD HL,(PC)
03220       CP #80
03230       JR C,PCAD4
03240       DEC H
03250 PCAD4 ADD A,L
03260       LD L,A
03270       JR NC,RT2
03280       INC H
03290 RT2   RET
03300 ;
03310 PRSPC LD A,#20
03320       RST #10
03330       DJNZ PRSPC
03340       RET
03350 PRINT LD D,A
03360       RRA
03370       RRA
03380       RRA
03390       RRA
03400       CALL P1
03410       LD A,D
03420       CALL P1
03430       RET
03440 P1    AND #0F
03450       ADD A,#30
03460       CP #3A
03470       JR C,P2
03480       ADD A,#07
03490 P2    RST #10
03500       RET
03510 ;
03520 TAB   LD A,2
03530       CALL #1601
03540       LD A,#D
03550       RST #10
03560       LD HL,(PC)
03570       LD A,H
03580       CALL PRINT
03590       LD A,L
03600       CALL PRINT
03610       LD A,"-
03620       RST #10
03630       LD B,8
03640 T1    LD A,#20
03650       RST #10
03660       LD A,(HL)
03670       CALL PRINT
03680       INC HL
03690       DJNZ T1
03700       RET
```

El corcho

Preguntas y respu

P No me quedó clara la forma de utilizar las variables de tres elementos como publicaron Uds. en el número 3 de TODOSPECTRUM en el programa *«Software para ordenar el software»*. ¿Qué significa cada uno de los elementos en la expresión DIN a$ (10,10,57) ¿Qué características técnicas tiene la impresora Epson FX-80? ¿Y el lápiz de luz de Dk'tronics? ¿Qué precio tienen?

Eduardo Martínez
Cartagena

R

*Con la sentencia a que te refieres DIMensionamos una tabla compuesta por 100 variables (10*10) de una longitud de 57 caracteres cada una. De esta forma quedarán a$ (1,1), a$ (1,2)... a$ (1,10), a$ (2,1)... hasta a$ (10,10). Añadiendo un tercer parámetro podremos fraccionar la cadena de la forma habitual. Por ejemplo, si queremos imprimir los 7 primeros caracteres de el primer elemento de la tabla deberemos hacer PRINT a$ (1,1, TO 7).*

En cuanto a las características de la impresora Epson FX-80 son las siguientes:

Representante en España: **Tradetec, Epson Center.** Provenza, 89-91. 08029 Barcelona. Tel.: (93) 322 03 54.

Características estándar: Velocidad: 160 CPS. Máx. caracteres/línea: 137. Ancho máx. de papel: 10".

Interfaces: Paralelo: *Centronics.* Opcional: Serie RS 232C.

Características particulares: Rollo de papel, *Buffer* de 2 KB extensible a 32 KB, impresión enfatizada, cursiva. Precio: 155.000 ptas.

En cuando al lápiz óptico Dk'tronicks *en este mismo número le dedicamos, como ya habrás visto, un artículo completo en el que podrás informarte exhaustivamente.*

P Poseo una impresora Admate-100 y me han avisado que no hace COPY a pesar del *interface Centronics.* ¿Se podría conseguir por *software*? ¿De qué manera?

Antonio García
Alicante

R

Sí es posible conseguir COPY en la impresora a que hace alusión, para ello puedes utilizar el programa Basic adjunto (programa 1); esta rutina puedes incluirla en tus programas mediante MERGE, y acceder a ella con GO SUB 9000. Debe funcionarte a la perfección, aunque, como quizá te resulte algo lenta, puedes recurrir al código máquina. Si te decides a ello el otro programa que te ofrecemos (programa 2) es una rutina en assembler *que puede solucionar el problema, pero, dependiendo del tipo de* interface *que tengas, es posible que tengas que realizar alguna modificación. Por si no tienes ensamblador incluimos también un programa Basic que se encargará de cargar el código máquina. Una vez ejecutado, y si no ha habido problemas en la suma de control, deberás salvarlo en cinta con SAVE «copy» CODE 65205, 100. Para volver a cargarlo deberás hacer LOAD «copy» CODE.*

```
8995 REM    programa1
8996
8997 REM  subrutina COPY
8999
9000 LPRINT CHR$ 27;CHR$ 27;"A";
CHR$ 27;CHR$ 8;
9010 FOR y=175 TO 7 STEP -8
9020 LPRINT CHR$ 27;CHR$ 27;"K";
CHR$ 27;CHR$ 0;CHR$ 27;CHR$ 1;
9025 FOR x=0 TO 255
9030 LPRINT CHR$ 27;CHR$ (128*PO
INT (x,y)+64*POINT (x,y-1)+32*PO
INT (x,y-2)+16*POINT (x,y-3)+8*P
OINT (x,y-4)+4*POINT (x,y-5)+2*P
OINT (x,y-6)+POINT (x,y-7));
9040 NEXT x: LPRINT
9050 NEXT y
9060 RETURN
```

```
  5 ;        programa 2
  6
  7 ;       subrutina COPY
  8
 10          ORG   65205
 20          LD    A,27
 30          RST   16
 40          LD    A,27
 50          RST   16
 60          LD    A,"A"
 70          RST   16
 80          LD    A,27
 90          RST   16
100          LD    A,8
110          RST   16
120          LD    HL,4000H
130 NULI     LD    A,H
140          CP    58H
150          RET   Z
160          LD    A,13
170          RST   16
180          LD    A,27
190          RST   16
200          LD    A,27
210          RST   16
220          LD    A,"K"
230          RST   16
240          LD    A,27
250          RST   16
260          LD    A,0
270          RST   16
280          LD    A,27
290          RST   16
300          LD    A,1
310          RST   16
320 NOLI     LD    B,8
330 NUCA     PUSH  BC
340          LD    B,8
350          LD    DE,TABLA
360 NUBY     RLC   (HL)
370          EX    DE,HL
380          RL    (HL)
390          INC   HL
400          EX    DE,HL
410          DJNZ  NUBY
420          INC   H
430          POP   BC
440          DJNZ  NUCA
450          EX    DE,HL
460          LD    B,8
470          LD    HL,TABLA
480 IMPR     LD    A,27
490          RST   16
500          LD    A,(HL)
510          RST   16
520          INC   HL
530          DJNZ  IMPR
540          EX    DE,HL
550          LD    C,L
560          INC   L
570          JR    Z,NULI
580          LD    A,H
590          SUB   8
600          LD    H,A
610          LD    A,C
620          AND   1FH
630          CP    1FH
640          JR    NZ,NOLI
650          JR    NULI
660 TABLA    DEFS  8
```

Y en el próximo número...



Preguntas y respuestas

```
  10 REM   cargador programa 2
  20 LET c=0
  30 FOR n=0 TO 99
  40 READ a
  50 POKE 65205+n,a
  60 LET c=c+a
  70 NEXT n
  80 PRINT  FLASH 1;" ERROR EN D
ATA " AND c<>10804
  90 DATA 62,27,215,62,27,215
 100 DATA 62,65,215,62,27,215
 110 DATA 62,8,215,33,0,64
 120 DATA 124,254,88,200,62,13
 130 DATA 215,62,27,215,62,27
 140 DATA 215,62,75,215,62,27
 150 DATA 215,62,0,215,62,27
 160 DATA 215,62,1,215,6,8
 170 DATA 197,6,8,17,24,255
 180 DATA 203,6,235,203,22,35
 190 DATA 235,16,247,36,193,16
 200 DATA 237,235,6,8,33,24
 210 DATA 255,62,27,215,126,215
 220 DATA 35,16,248,235,77,44
 230 DATA 40,188,124,214,8,103
 240 DATA 121,230,31,254,31,32
 250 DATA 205,24,175,243
```

Como podrán observar quienes tengan algún conocimiento del sistema operativo del Spectrum, el programa en código máquina utiliza el RST 16 (10H) para imprimir un byte. *El programa supone que previamente se ha seleccionado como canal de salida el canal de impresora (canal 3) y que la rutina a la que apunta este canal necesita un ESC (27) antes de cada código de control. Para utilizar el programa, este debe ir precedido de una llamada a la rutina de selección con un 3 en el acumulador:*

LD A,3

CALL 1601H

El programa está preparado asimismo para un Interface *de impresora tipo Kempston (modelo 1). Este* Interface *requería un código ESC antes de cada caracter de control que debiera ir a la impresora. Para utilizar el programa con, por ejemplo, el* Interface *1, basta abrir el canal 3 a «B», incluir al principio las dos líneas que hemos indicado anteriormente, y eliminar*

LD A,27

RST 16

una vez cada vez que ocurra dos veces seguidas en el programa. También se puede «pokear» con NOP.

El programa se puede relocalizar a cualquier posición de memoria, ya que la única dirección absoluta es TABLA. Por ello, basta asignar hueco para una tabla de 8 bytes.

DIRECTOR:
Simeón Cruz
COORDINADOR EDITORIAL:
Emiliano Juárez
REDACCION:
Juan Arencibia, Fernando García, José C. Tomás, Luis M. Brugarolas, Ricardo García, Santiago Gala
DISEÑO: Ricardo Segura

Editado por
PUBLINFORMATICA, S. A.
Presidente: Fernando Bolín
Director Editorial: Norberto Gallego

Administración:
INFODIS, S. A.
Gerente de Circulación y ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González. Tel. 733 79 69
Administración:
Miguel Atance y Antonio Torres
Jefe de Publicidad
María José Martín
Dirección y redacción:
Bravo Murillo, 377-5.° A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
Administración y Publicidad:
Bravo Murillo, 377-3 E. Tels. 733 96 62/96
Publicidad Madrid:
María Jose Martín
Publicidad Barcelona:
María del Carmen Ríos, Olga Martorell. Pelayo, 12.
Tel. (93) 318 02 89.
08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.

Distribuidor en VENEZUELA,
SIPAM, S.A.
AVD. REPUBLICA DOMINICANA.
EDIF. FELTREC – OFICINA 4B
BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la Asociación de Revistas de Información ARI asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.°
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.° A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo 377, 5.° A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.


Programas/Juegos/Montajes/Código Máquina
Año II N.º 21 Agosto 1985-250 Ptas.
ZX
REVISTA PARA LOS USUARIOS DE ORDENADORES SINCLAIR
Resuelve el crucianagrama y gana un Invesdisk
YA ESTA A LA VENTA
MAPA DEL LABERINTO
Underwurlde, lo mejor de Ultimate
NOVEDADES CALIENTES
Gyron y Shadow Fire
Selección de libros para el verano